AD_____

GRANT NUMBER DAMD17-94-J-4486

TITLE: MediSIM: Simulated Medical Corpsmen for Medical Forces Planning and Training

PRINCIPAL INVESTIGATOR: Dr. Norman I. Badler, Bond-Jay Ting, Francisco Azuola, Pei-Hwa Ho, Suejung Huh, Evangelos Kokkevis

CONTRACTING ORGANIZATION: University of Pennsylvania
Philadelphia, Pennsylvania 19104-6145

REPORT DATE: October 1996

TYPE OF REPORT: Annual

PREPARED FOR: Commander
U.S. Army Medical Research and Materiel Command
Fort Detrick, Frederick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for public release;
distribution unlimited

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1996	Annual (8 Sep 95 - 7 Sep 96)

4. TITLE AND SUBTITLE
MediSIM: Simulated Medical Corpsmen for Medical Forces Planning and Training

5. FUNDING NUMBERS
DAMD17-94-J-4486

6. AUTHOR(S)
Dr. Norman I. Badler, Bond-Jay Ting, Francisco Azuola, Pei-Hwa Ho, Suejung Huh, Evangelos Kokkevis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Pennsylvania
Philadelphia, Pennsylvania 19104-6145

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Commander
U.S. Army Medical Research and Materiel Command
Fort Detrick, Frederick, MD 21702-5012

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200*
This document presents the details of the construction of the Jack human figure model. It explains the methods used in putting the model together, the data and references employed, as well as other related topics regarding the application of this model within the interactive 3-D environment of Jack.

14. SUBJECT TERMS MediSIM, Computer and Information Science

15. NUMBER OF PAGES 144

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

FOREWORD

Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the U.S. Army.

( ) Where copyrighted material is quoted, permission has been obtained to use such material.

( ) Where material from documents designated for limited distribution is quoted, permission has been obtained to use the material.

( ) Citations of commercial organizations and trade names in this report do not constitute an official Department of the Army endorsement or approval of the products or services of these organizations.

( ) In conducting research using animals, the investigator(s) adhered to the "Guide for the Care and Use of Laboratory Animals," prepared by the Committee on Care and Use of Laboratory Animals of the Institute of Laboratory Animal Resources, National Research Council (NIH Publication No. 86-23, Revised 1985).

( ) For the protection of human subjects, the investigator(s) have adhered to policies of applicable Federal Law 32 CFR 219 and 45 CFR 46.

( ) In conducting research utilizing recombinant DNA technology, the investigator(s) adhered to current guidelines promulgated by the National Institutes of Health.

Norman L Badler
Principal Investigator's Signature

10-18-1996
Date

Contents

List of Figures

Chapter 1

Introduction

This document presents a detailed record of the methodologies, assumptions, limitations, and references used in creating the human figure model in *Jack*. The model itself is a work in progress and many issues regarding its performance will continue to be improved. It is the understanding of the authors, and should be for the users of this model as well, that there will be constant changes to the human model as more data becomes available, more research comes to fruition, and more powerful hardware is available. This document will continue to be updated to reflect those changes.

The purpose of having a three dimensional articulated human model in a graphical environment is to be able to simulate human tasks and functions in a virtual environment before a real world mock-up is actually built. In order to fulfill such goals, the human model must be built so that it reflects specific characteristics of real humans. In this particular context, the model must have comparable joint centers and range of motions, and body dimensions, for example, to those of its real-world counterpart. It is this goal that the *Jack* human model is attempting to achieve.

1.1 The Human Model

The human model in *Jack* has gone through many changes over the past years, from the earlier skinny body [25] where body segments were represented by tetrahedrons of various sizes, to the current (under-construction) body with realistic (human-like) geometric shapes. As more and more details are put into the human model, more data are needed to support this effort. The model is subject to change when more powerful hardware and algorithms become available and when more data are collected.

The human model is composed of segments connected together by joints. A segment usually represents a distinguishable part of the body, like the head. The shape of the model is represented by the geometries of the segment. For example, the geometry of the head is associated with the head segment. The articulation is formed by defining joints between segments, e.g. the elbow joint is the articulation between the upper arm and lower

arm segments. Each joint has an associated range of motion, and a number of degrees of freedom (DOF).

Jack uses the "Peabody" language syntax to describe figures (usually segmented, but not necessarily human) which are to be displayed or manipulated. A Peabody description file allows for the definition of joints, constraints, and sites (attachment points) for the figures. The human figure model topological structure is defined using such a file, which, for the case or human figures, we refer to as the "human figure definition file". The human figure definition file contains other information besides the segment dimensions (geometric surface scale factors). For example, for segments, one finds:

- segment geometric surface and its associated scaling factors
- color attributes
- segment mass
- segment center of mass
- segment sites (location and orientation)

and for joints,

- connect "siteA" to "siteB"
- type (degrees of freedom and orientation)
- lower and upper limits for each degree of freedom
- displacements (if any).

1.2 Anthropometry-Based Human Model Construction

This section addresses assumptions and limitations in building the model which are mostly due to, first, insufficient data, and second, the lack of an "absolute" analytic or mathematic model.

1.2.1 Joint Center Locations

The location of joint centers, which collectively define the model link structure, is estimated based on a biostereometric human body sample. No anthropometric survey to date provides joint center location data. We acknowledge that the quality of such estimation is limited. Further work in this area is needed.

1.2.2 Independent Degrees of Freedom

The range of motion specified in our model came from studies conducted by NASA [38]. Each degree of freedom is measured independently. In reality, the DOF's are not independent of each other, either within the same joint or sometimes across joints, due to the structure of the joints and the muscle groups that control them. Thus, it is possible that the figure model may accept to be positioned in a configuration that may appear awkward or unrealistic.

1.2.3 Floating Anthropometric Measurements

The current data from standard anthropometric surveys are collected with devices such as tapes, calipers, and anthropometers [28]. These measurements are numerical values, many of which do not refer to a fixed reference in three dimensional space. When such data are used to build 3D human models, it is left to the figure model designer to decide where and how it is employed.

1.2.4 Model Geometries and Dimensions

An anthropometry-based human model needs to have built-in support for scaling so that models of different dimensions can be created. The various $3D$ shapes of the human body segments require a more detailed description than the $1D$ anthropometric data can provide, and thus, when scaling is applied to change the dimensions of a model, certain assumptions are made about the geometries to compensate for insufficient data. An example of these assumptions is the adoption of more or less regular shapes to model segments, thus reducing the number of parameters needed to control the geometry's size and shape. The polygonal model, Polybody, uses linear, bounding box scaling for all geometries except the arms and fingers, in which tapering scaling is used.

A smooth-skinned body is now available where the geometries remain continuous across joints when joint angles are changed. It uses realistically shaped geometries to better represent the human in $3D$. More sophisticated scaling methods are under development that would provide higher levels of control of body shapes, which can be used in conjunction with the smooth-skinned body to size it to different dimensions.

1.3 Recent Developments

This section outlines major improvements and work related to the human model done within the past year.

Current efforts in the human figure modeling area focus on model improvements, statistical anthropometric data processing methods, enhanced human figure model construction and measuring methods, and automated accommodation analysis.

1.3.1 Cervical Joints

Work was completed to give the *Jack* human figure cervical vertebrae. The cervical joints can be manipulated similar to the seventeen-segment spine. There is a set of default degrees of freedom and ranges of motions, but the user is allowed to redefine them, or to "freeze out" certain portions of the cervical column, to model more restrictive neck movements under constraints or injuries. The inclusion of the cervical joints greatly enhanced the behaviors of the head, which now moves in a very realistic fashion due to the more realistic distribution of neck bending, twisting, and lateral bending.

1.3.2 Spine Curvature Improvements

The curvature of the spine of the previous model was almost non-existent. This contradicts what one finds in a real torso. The curvature is now consistent with that of a real-world average human, and, therefore, the flexible torso range of motion should be more realistic due to the improved locations of joint centers along the spine.

1.3.3 Visible Landmarks

Various utilities have been developed to define landmarks, place and move landmarks to arbitrary positions, color landmarks, label landmarks, and display landmark labels.

These utilities were developed with the understanding that, as the human model improves in its representation of a real human body in terms of a link structure and geometric segment shapes, anthropometric landmarks will play a crucial role in defining various properties of the model.

1.3.4 Automated Accommodation Analysis

An automated accommodation testing utility has been developed that includes any posture defined by the user. The utility takes a user-defined environment, which may contain multiple human figures, and performs the testing on all human figures in a user-specified directory. Each figure is placed in the same environment under the same human behavior control and constraints. The utility then evaluates all constraints and records the offset distances for them. Currently the utility can test up to five hundred figures at a time, at a rate of approximately ten seconds per figure on an Indigo 2 workstation with sixty-four megabyte of memories. The results of the testing are written to a plain ASCII file for post-processing and analysis.

This utility is very useful when evaluating a workspace design. Depends on the requirements of the designer a workspace may need to be evaluated with a couple, a hundred, or even thousands of figure models. Combined with the right figure generation algorithms this utility automates part of the human engineer's simulation tasks and provides useful quantitative data for detailed analysis.

A Monte Carlo based figure generator has been developed to supply the test simulation with the necessary test cases. In addition, a Cadre (Boundary) family generator has been developed to complement the Monte Carlo generator.

1.3.5 Improved Surface Geometry and Smooth Skin

With the improvements in hardware performance, more realistic, non-rigid geometries for the human model can be used.

A male model is constructed with realistic geometric shapes, and utilities are provided to deform the geometries so that, after a joint angle changes, the geometries around the joint remain smooth. The same technique is applied to the torso, neck, and hip so that the full body is smooth, except the hand and fingers.

1.3.6 Dynamics and Joint Torque Loading

Jack now has its own dynamics computation modules, which allows for the computation of joint torques when a load is attached to the human figure. This feature gives the users of *Jack* the ability to compute such loads interactively. It also facilitates the comparison of such computations with experimentally obtained data.

1.3.7 Anthropometric Methods

A figure model has been created following an object-oriented approach. The figure object consists of the data storage structures plus two main methods, namely, a constructor and a measuring method. All the rules used by both methods are (user-) redefinable, but defaults are provided. Having the construction and measuring rules exposed simplifies the task of validating, verifying, improving, and updating the model. The figure object's implementation has been highly optimized for efficient creation of large populations of figure instances.

1.4 Conclusions

To evaluate any anthropometry-based human model one must establish a suite of tests and measurements that can quantify a model's performance with respect to clearly defined tasks and goals. From an user's point of view such tests can be used to select a model best-suited for specific tasks. From a modeler's point of view they can identify limitations of a model and, perhaps, point to areas where improvements can be made.

The most basic test suit should include anthropometric measurements on the model against the survey from which the model is constructed. Out of the hundreds of traditional anthropometric measurements a subset should be selected that focuses on the part of the human model most relevant to the intended use or tasks.

The next level of tests should focus on the model's functional performance or task-specific analysis. Possible candidates for this tests are functional reaches and reach space for specific joint chain. These test could reveal the model's validity in placement of joint centers, joint limits, distribution of joint displacement in different postures , and control of multiple joints.

Another test should be designed to test the dynamic capabilities of the models. Such test should compare the computed forces and torques in the model against experimentally obtained data.

The interpretation and analysis of test results should be based on multiple tests on various sized models generated from the same modeling system with specific figure generation schemes, e.g. Monte Carlo. Statistics should be compiled for each modeling systems so that meaningful, task-specific, or posture-specific comparisons can be made.

1.5 Other Important References

To trace the *Jack* human model developments and evolutions, interested parties can refer to the following documents for detailed information. Also the web site, http://www.cis.upenn.edu/~hms/jack.html can be checked for latest announcements.

- Anthropometry for Computer Graphics Human Figures [26].
- 1988 Anthropometric Survey of U.S. Army Personnel: Methods and Summary Statistics [22].
- SASS v.2.5 User's Manual [4].
- Jack 5 User's Guide [43].

1.6 Overview of Chapters

Chapter 2 discusses the details of building human models in general, followed by discussions on model anthropometry in Chapter 3, and of scaling in Chapter 4. The details about the *Jack* anthropometry-based human models are presented in Chapter 5. Chapter 6 presents the details concerning the statistical data processing methods. Chapter 7 discusses the figure generation tools that we developed and Chapter 8 discusses anthropometric errors. Inverse dynamics is discussed in Chapter 9 and Chapter 10 discusses the details of the smooth-skin implementation. Chapter 11 discusses methods used in estimating landmark locations on the model.

Chapter 2

Virtual Human Models

This chapter describes the construction of virtual human analogues that model the structure of real humans. Such analogues are used to simulate human functions in a virtual environment.

We start by examining the human body structure, followed by a review of issues in building such models.

2.1 The Human Body

Anatomically, the human body is composed of the skeletal system, the muscular system, the inner organs, and the skin and fat layers [8, 49]. The skeletal system and the muscular system, jointly called the musculoskeletal system, protect, support, and move the body, while the organs maintain a stable internal environment (homeostasis). The overall shape of the human body is determined mostly by the amount of fat and muscle it has and how they are distributed. The size of the body is influenced by the skeleton and the soft tissues that cover it.

The human skeletal system is composed of over two hundred bones. Bones are connected together through joints, and, collectively, they form the articular system. It is this system that determines the maximum amount of flexibility, mobility, and range of movements for the body. Such limits are affected by many other factors like the amount of fat, clothing, equipment attached to the body, age, or physical health. The skeleton's size, structure, and proportions are affected by age, exercise, genetics, nutrition, and even profession [46] but interest is focused in modeling the skeleton of an average, healthy adult.

Joints in the human body are very complicated. Some types of joints, like the fibrous and cartilaginous joints, allow little, if any, movements while others, like the synovial joints, allow specific types of movements. Synovial joints are thus the most important joints when it comes to human posture and motions [8].

Synovial joints can be further classified into uni-axial (hinge), bi-axial (condyloid), and multi-axial (plane, ball-and-socket, saddle, pivot) joints [8].

Muscles provide the needed forces for the body to support itself, move, and posture. Muscles often work in groups, e.g. the biceps and triceps to control the elbow [8]. The correspondence between joint and muscles is not only one to one or one to two but can be one to many and many to one. Take the spine for example, it has thirty-four vertebrae and thirty-five joints and more than twenty muscle groups to support its movements and postures [8, 35]. The shoulder is another example with multiple muscles jointly controlling multiple joints [51, 39, 27].

On top of the skeleton and muscles, there are soft tissues and skin that form the final shape of the human body. The amount of muscles and fat tissues are the two dominant factors in shaping the human body [7]. They are affected by many factors, genetics, exercise and dietary habit, to name a few [7].

2.2 Virtual Human Models

A virtual human models only a subset of the anatomical and biomechanical aspects of a real human, depending on the purpose and level of abstraction. Interest is put on the construction of a scalable model that represents an average human in size and possesses similar flexibility and range of movements. This requires having realistic geometric shapes, an acceptable representation of the articular system, and proper sizing algorithms.

2.2.1 Joints and Joint Centers

For the model to behave properly the joint locations should be specified appropriately within the human body segments. Also, it is necessary to specify the way they rotate or translate about the correct axes with proper limits. When the geometries and links are sized to different dimensions it is important that the characteristics of the articular system are passed along into the new (size) figure instance. Fig. 2.1 shows one of our polygonal human models, Polybody [44].

If all the reachable points for the end effector of a single joint are lumped together as a reach space [1], then incorrectly placed joint center or joint axis will cause a shift or rotation of such reach space relative to the correct joint center, while incorrect joint limits will enlarge or reduce such reach space [28]. When more than one joint is involved, errors in modeling joints affect both the size, shape, location, and orientation of such reach space.

Techniques developed in the field of robotics [41, 50] are often used for modeling and controlling joints in virtual human or character models. When "robot joints" are used to model joints in a human model, its articulation can be uniquely defined by its link

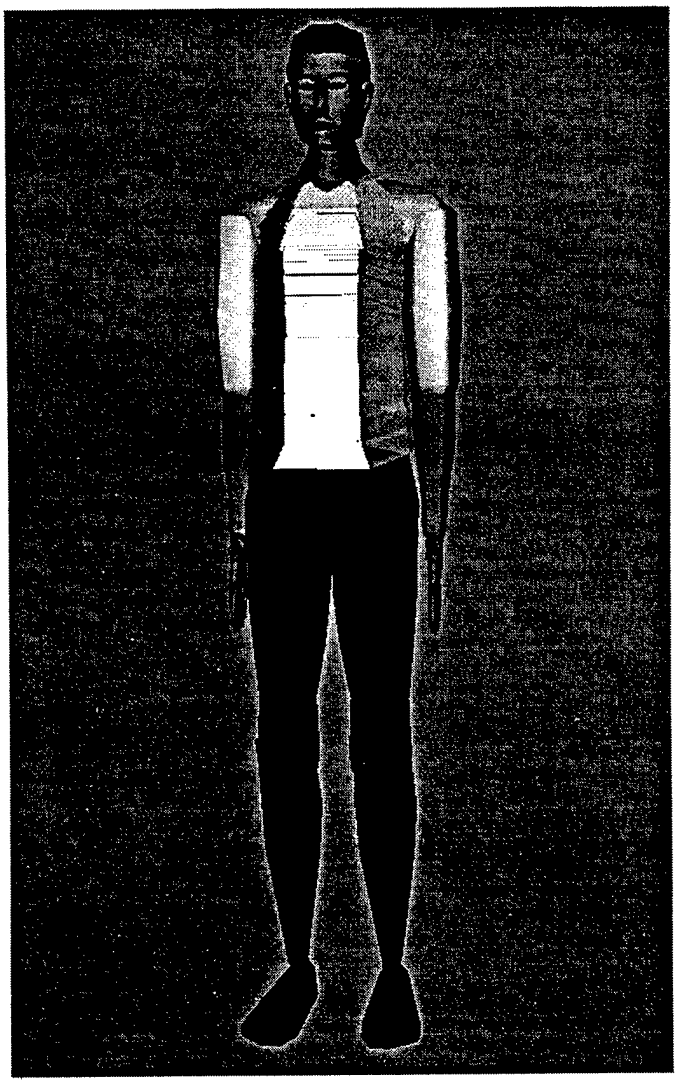

Figure 2.1: Polybody Human Figure Model

mechanism, which states the linkages (joints) among the segments (links). Linkages are specified within each segment's local coordinate frame [44, 41]. To go from one segment to its neighbors one simply performs the geometric transformations associated with the sites that define the joint.

Since the joints in the human body usually do not have perfect spherical shape and seldom rotate around a fixed point or plane, additional errors are introduced if "robot joints" are used. This is an area where better joint models require more computation and thus adversely affect system performance. A decision has to be made between better performance or better accuracy. Sometimes, errors from other sources may be much more serious than modeling errors, and, therefore, "robot joints" prove to be not a bad choice [51].

2.2.2 Model Geometry

Geometries for a human model can be obtained in several ways. Artistically, one creates the geometries that may or may not resemble a real human. The level of detail and realism we can achieve depend upon the skills of the modeler (artist). The polygonal human model in Fig. 2.1 is composed of about 2400 simple polygons. It has a rather artificial appearance. Geometric shapes can also be obtained by scanning real human subjects with the help of imaging or photographic equipment [18]. The level of detail can be extremely high yielding very realistic representations. Fig. 2.2 is the human model from Viewpoint Datalabs, created by medical illustrators. It has close to 40,000 polygons. In general, human body geometric shapes are available in all levels of detail.

Depending on the source of the geometry, we may or may not know all the associated physical and physiological attributes. Attributes related to the geometry itself like area, distance, or volume can mostly be estimated numerically, but those related to the physiological properties of human bodies can only be obtained from real human data.

2.2.3 Control and Manipulation

Once a human model is constructed, it can be used for a wide range of interests, from entertainment to simulation. The intended purpose of our research is to provide realistic 3D human models for human factor engineers, to facilitate their evaluation of workspace designs. For this purpose, utilities need to be developed so that the human model can be easily put into various postures where reach, field of view, and space accommodation analyses can be conducted [6, 42]. Without realistic human models, however, little value can be generated with such tools. It is our goal to generate figure instances that are realistic, not only in shape, but in their anatomical, biomechanical, and anthropometrical configurations, so that these figure instances prove to be useful when applied in those analyses.

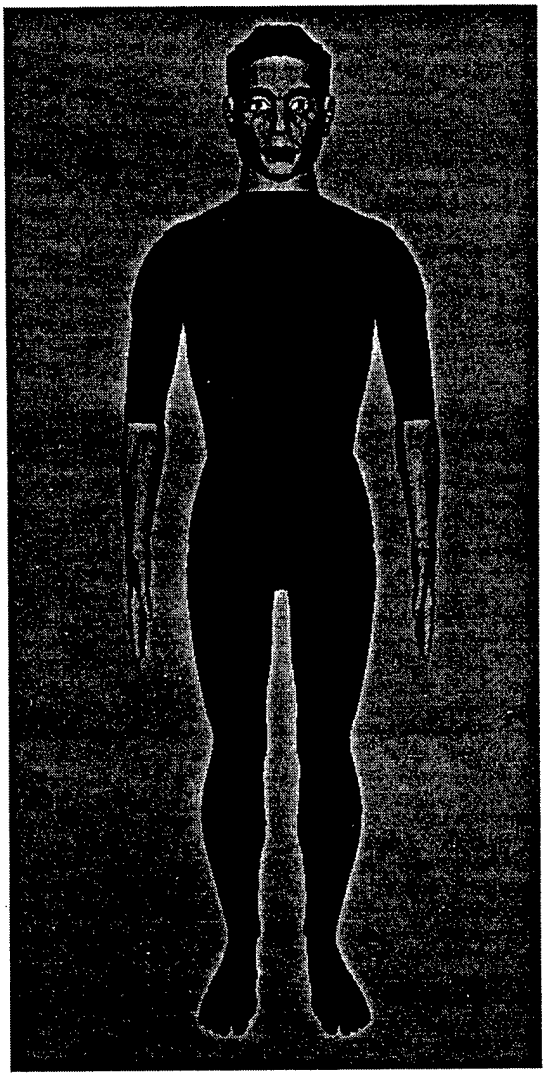

Figure 2.2: Viewpoint Human Figure Model

Chapter 3

Human Figure Model Anthropometry

In evaluating a workspace design one needs not just one human model but many models of different sizes. To have models of various sizes means, first, that there are mechanisms to control and measure the size of the model, and, secondly, that there are established sources of human data to support the sizing operations. For the latter case, one finds many anthropometric surveys providing data sets on populations of human subjects. We address measurements of the model in this chapter. Chapter 4 addresses the issues in scaling the human model and Chapter 5 discusses how the 1988 Anthropometric Survey of U.S. Army Personnel is used to generate the Jack figure models.

3.1 Measurements on the Human Model

Human models can only be sized if the respective parameters, or attributes, can be computed (measured) on the model. For instance, segment length can be used if one can compute segment length on the model. As another example, density cannot be computed from the model's associated geometry, but it is artificially associated with it. Hence, the density parameter can be used and changed in a model, even though no physical, or visual, changes can be associated with such parameter, unless it affects a visible attribute such as volume.

3.1.1 Distance

Distance is an attribute that can be computed exactly between two $3D$ points in virtual space. It is associated with attributes like length, width, and depth.

3.1.2 Circumference

Circumference can not be computed exactly in models since the geometries used are an approximate representation of the real human geometric shapes. Circumference must be defined relative to a fixed plane or a cross-section, but the population surveys do not provide support information, which leads to having differences in such planes between the real human and the model, which introduce error into the model.

3.1.3 Volume

Volume is another attribute that can only be viewed as an approximation of a real human segment volume. When density is used in a human model the mass of that model can be computed once volume is obtained. Between volume, density, and mass there are only two free parameters, the third one can be derived from the other two.

3.2 Significance of Measurements

It is important that the measurements on human models are viewed with the limitation of such models in mind. Circumference, for example, is affected by the plane on which it is computed and by the realism of the model geometries. A model can have good distance accuracies if it is a good representation of the human body in terms of link length and joint center locations. Circumference and volume, for example, are more difficult to represent since they require accuracy to be taken care of in more than one dimension.

Chapter 4

Scaling Geometric Objects

In this chapter we discuss issues in scaling geometric objects. Techniques that can be applied to the Jack's scalable models are presented.

4.1 Scaling Defined

Scaling is a transformation process that changes a geometry's dimensions, shape, or both. The process is not affine [21] since it does not preserve shape or angles. We differentiate dimensional scaling from shape control. The former emphasizes on controlling the dimensions of geometries and the latter focuses on non-quantitative means to control shapes of geometries. The two are not independent issues since scaling can change both dimensions and shapes.

Scaling can be defined as a function $S(x, y, z)$ applied to a geometry $G(x, y, z)$ to obtain a new geometry $Gnew(x, y, z)$.

$$Gnew(x, y, z) = S(x, y, z) \cdot G(x, y, z). \tag{4.1}$$

4.2 Dimensional Scaling

Dimensional scaling of geometries transforms geometries so that they satisfy specific dimensions (thus the name dimensional scaling). For the anthropometry-based human models the type of specifications include measurements of body segment length, depth, and width.

Depending on the specifications different scaling approaches may be required. We can start with linear scaling and then move onto more advanced scaling techniques to support more flexible specifications.

4.3 Normalization

To prepare a geometry for scaling a coordinate frame has to be chosen first. Oftentimes the geometry is scaled down to unit dimensions for subsequent scalings. This process is called normalization. It is equivalent to placing the geometry into a bounding box centered around the origin of the coordinate frame. Once a geometry is normalized it can be scaled by specifying how the bounding box is stretched by various functions, constant, linear, non-linear, etc. Fig. 4.1 shows a human head model before(left) and after(right) normalization with the bounding box shrank to a two by two by one cube.

4.4 Linear Scaling

Linear scaling is the simplest type of scaling. A geometry is simply stretched or shrank uniformly in each dimension. This technique is compatible with normalized geometries. This is also called a bounding box approach since the process simply scales the bounding box uniformly with the geometry in the box.
Linearly scaling can be defined by:

$$Gnew_x = Sx \cdot G_x,$$
$$Gnew_y = Sy \cdot G_y,$$
$$Gnew_z = Sz \cdot G_z.$$

where Sx, Sy, and Sz are constants.

With linear scaling only specifications that are related to the three orthogonal dimensions can be used, i.e. length, width, and thickness. Circumference can be used as well, but, since it is related to two dimensions that define the plane where it is taken, given the circumference value, one is left with an additional degree of freedom for the scaling algorithm.

Provided that the reference frame is set up correctly the bounding box approach is suitable only for symmetric objects like cylinders and ellipsoids. These objects have their axes of symmetry at the center of the object and coincide with the axes of their coordinate frames. Objects that are not symmetric or irregular, like human body segments, don't scale well with this approach.

4.5 Extensions of Linear Scaling

One simple extension of linear scaling is what we called tapering scaling where different scale factors are assigned to different nodes along the same axis. The underlying geometry gets scaled by the exact scale factors at the nodes, and, in-between nodes, a linearly interpolated scale factor is applied. This is a technique that can be used to scale object

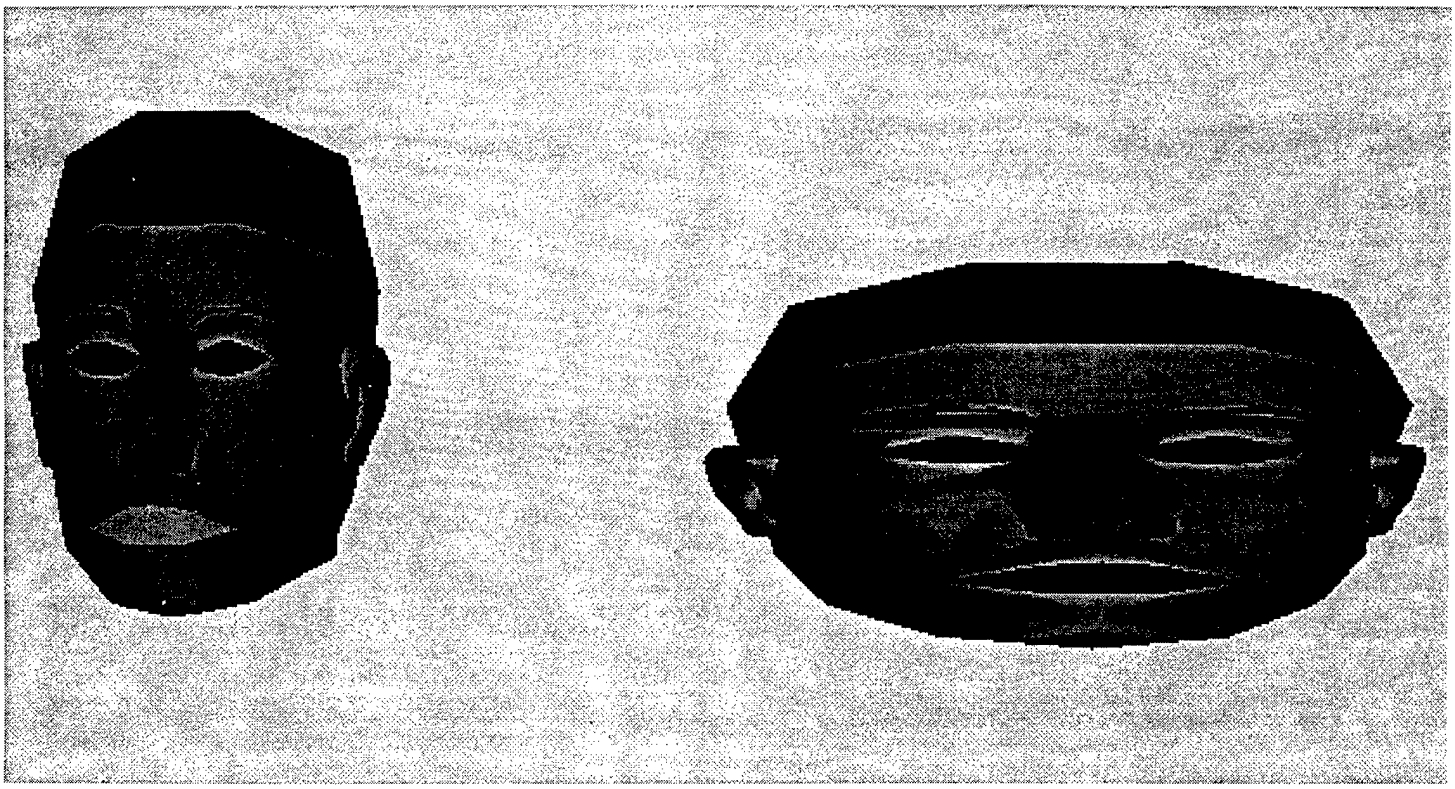

Figure 4.1: Original and Normalized (Scaled Up 16 Times) Head Model

that has different scaling specifications at various points. Good example of cases that may benefit from this type of scaling are the forearm and thigh segments, where the widths and depths at the two ends may need to be scaled differently.

We can define segmented scaling, assuming that the geometry is segmented into n regions along the z axis by nodes $z_0, z_2...zn$, as follows:

$$Gnew_x = \begin{cases} Sx_i \cdot G_x & \text{for } z = z_i \\ Interp_x(z) & \text{for } z_i > z > z_{i-1} \end{cases}$$

$$Gnew_y = \begin{cases} Sy_i \cdot G_y & \text{for } z = z_i \\ Interp_y(z) & \text{for } z_i > z > z_{i-1} \end{cases}$$

$$Gnew_z = G_z.$$

for $i = 0, 1, 2..n$ and $Interp$ is a linear interpolation function [21] defined as:

$$Interp_x(z) = \frac{z_i - z}{z_i - z_{i-1}} Sx_{i-1} + \frac{z - z_{i-1}}{z_i - z_{i-1}} Sx_i \qquad \text{for } z_i > z > z_{i-1}$$

$Interp_y$ can be defined similarly.

This method provides better shape control compared to the bounding box approach but only works well with objects that are symmetrical at each cross section along the long

axis.

4.6 Non-Uniform Scaling

To overcome the constraints imposed by linear and tapering scalings non-uniform and non-linear scalings are used.

Fig 4.3 shows the original and non-uniform scaled leg of a human model. The scaling function used is a combination of sinusoidal functions, *sine* and *cosine*, and linear interpolation. The scaled geometry has a different shape and continuity is maintained.

It is clear that there are unlimited ways to transform the size and shape of a geometry. The important issue is how to guide and determine the techniques needed to efficiently morph a realistic human model into another (realistic) one with different anthropometric dimensions.

4.7 Supported Specifications

With linear scaling length, width, and depths are directly supported. The disadvantage of using those specifications is that the resulting geometries are "boxy" and that graphical continuity may not be preserved since two neighboring geometries may have different specifications. Fig 4.2 shows a model leg that has discontinuities at the knee due to linear scaling. To ensure continuity one would require that all neighboring geometries have the same scale factors at their plane of intersection with their neighbors. Segmented scaling allows the continuity to be maintained, as long as scale factors are the same at the junctions. This is further supported by the fact that there is anthropometric data taken at segment junctions available.

The list of specifications that are used to describe human bodies is limited, even though not necessarily satisfied from an anthropometric survey [28, 22]. More realistic scaling of human models requires better scaling algorithms as well as better survey data [34, 28].

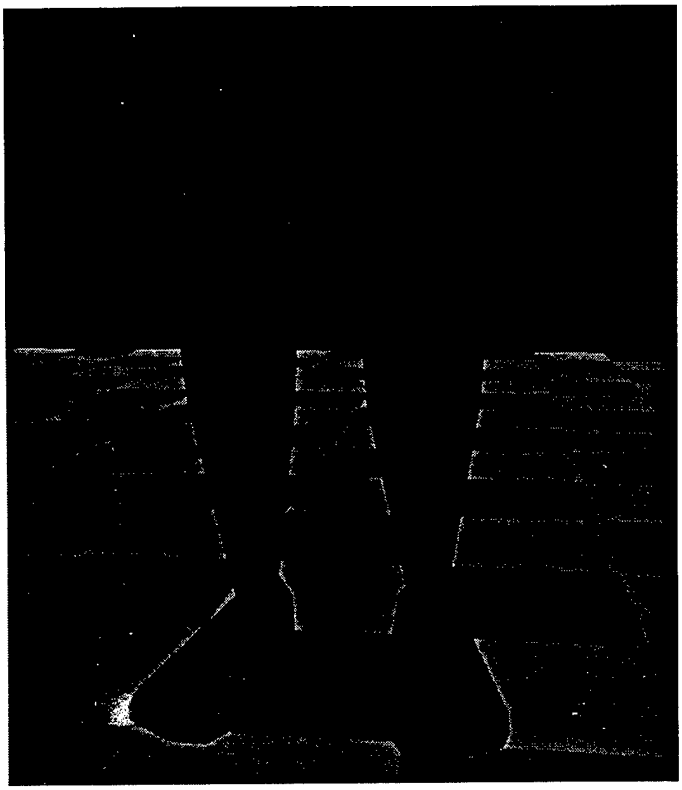

Figure 4.2: Anomalies caused by linear scaling.

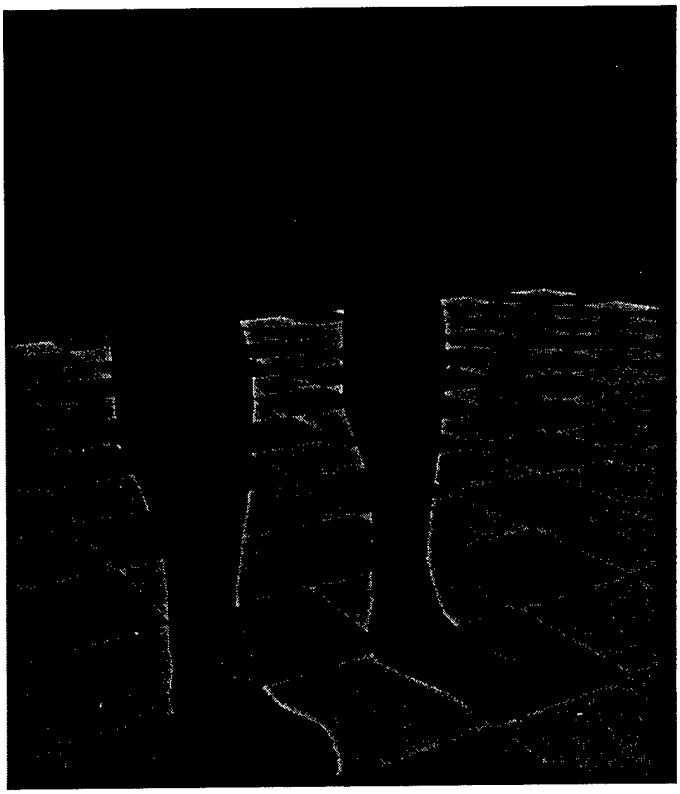

Figure 4.3: Results of non-uniform scaling.

Chapter 5

Anthropometry-Based Human Model Generation

5.1 Introduction

In this chapter, an overview of the aspects associated with the construction of the Jack's figure model is presented.

The figure model contains two main sets of capabilities. The first is the figure building capabilities set. The second is the set of anthropometric measuring capabilities. Statistical capabilities for data pre-processing are described in the next chapter.

5.2 Figure Model Methods

A figure model may be implemented as an "object". An object, in simple terms, is a collection of capabilities, plus the data associated with the implementation of those capabilities. From an anthropometric standpoint, a figure model has two major sets of capabilities (the implementation of which we refer to as methods), namely a set of figure building methods and a set of anthropometric mensuration methods. Each of these sets is considered next.

5.2.1 Figure Building Methods

Anthropometric figure building may be modeled as a mapping between data and figure model. There is not a one-to-one mapping between anthropometric data and a geometric human figure model. The mapping is, in practice, a composition of transformations such as regression computations, combination of measurements, geometric scaling operations, etc. We refer to all the operations performed by this mapping as figure building methods.

The figure building methods provide the necessary capabilities to produce a figure instance sized according to a given set of (input) dimensions. To simplify the creation process (from a user point of view), a constructor method is provided. The constructor

method can be seen as being a "master" method that takes anthropometric data as input and generates a figure object instance.

The figure building methods are specific of a given figure model. Based on the type and number of geometric segments, the type and number of links, and the landmarks and joint centers defined in the model, the methods perform the job of constructing a figure out of these components.

However, it must be noticed that these methods store no knowledge regarding the proportionality (preservation) aspects of the scaling. This information is assumed to be part of the input data. That is, all necessary statistical processing of the anthropometric data, including regression computations, estimations, etc., are assumed to take place as a step preceding the construction of the actual figure instance. The constructor takes this pre-processed data, along with additional knowledge about anatomical aspects of the human body, as well as geometrical assumptions such as symmetry, and produces a figure instance.

One of the major problems found in traditional anthropometric models is that all the methods of the figure model are embedded inside a program. This approach suffers from serious problems. For example, if a component of the figure model changes, it, usually, results in the need for major overhauls of the program. Another deficiency of this approach is that the methods of the model are fixed and hidden from the user. If the user needed to adapt or tune up the methods for a particular application, it would not be possible.

From an object-oriented point of view, the object's member functions are defined at the object's definition time. While the member functions of the object may remain fixed throughout the life of the object, an object-oriented framework allows for redefinition of such member functions, i.e., a (new) member function taking the place of a default one.

From a figure model standpoint, one would benefit from an implementation of such O-O framework, and even more if one considers an open architecture approach. The idea behind such approach is simple. A figure model is implemented as a "dumb" object, with no geometry, no link structure, no methods, i.e., nothing attached to it, except for a constructor method. The constructor method provides a way to start the building of figure object instances process. However, at instantiation time, the constructor expects all the necessary information and building methods to be supplied to it, in order to be able to construct the desired figure instance.

5.2.1.1 The methods

In Jack, the figure model constructor takes the form of a stand-alone piece of software. The constructor module is available also through the shell program *Genfig*, which will be introduced later in this document.

The definition of the figure model topological structure is stored in a figure definition file. This file contains all the details regarding, for instance, segments, articulations, attributes, etc. The figure definition file must be designed before any figure creation operation takes place. A default file is provided for the user's convenience. Specifications of an initial three-dimensional topological connection are provided. The figure file contains

what we refer to as a "generic" figure. From a geometric standpoint, it is complete in the sense that it can be displayed on a simulated environment created in the Jack GUI. From an anthropometric point of view, it serves the purpose of defining the figure elements that are to be manipulated by the anthropometric building methods.

The task of the anthropometric building methods is to embed the anthropometric data into the generic figure. The constructor contains a set of primitive auxiliary methods whose purpose is to interpret rules. The collection of primitive auxiliary methods and interpreted rules define the building methods. From a user standpoint, the primitive auxiliary methods are hidden in the figure constructor. The only thing that she needs to be concerned about is in defining the rules.

The rules have been divided in two parts, according to an intermediate and target data representations abstraction. This abstraction helps isolate the purely anthropometric aspects of the data from the geometrical particularities of the figure model.

The intermediate data representation is basically a definition of anthropometric variables in terms of formulae based on ANSUR 88 measurements. The variable names are arbitrary, even though, they are meant to remind us of the specific body segment which they represent, from an anatomical point of view. Also, the axes associated with each definition are defined to denote breadth, depth, and length (x,y,z, respectively). Figs. 5.1 and 5.2 show a summary of the default version of this representation. The formulae are arithmetic expressions based on ANSUR-88 variables. The variable numbers[1] appear between square brackets. In other words, a number between square brackets is meant to be taken as a variable number, and not literally. Also, some formulae of certain variables are commented as "guestimated" to indicate that the variable's value is computed approximately using a guessed factor[2]. Notice that all these formulae are user re-definable.

The target representation of the rules uses the variables defined in the intermediate data representation to define variables associated with the (target) figure model. The latter variables are the ones used to, ultimately, specify the scaling of the figure model.

Figs. 5.3 through 5.8 present the target data representation. The following aspects must be noticed. First, there is a reoccurring 0.5 scaling factor in 2 of the axes. The reason for the use of this factor is arbitrary and has to do with the way geometry is normalized (see Geometry Normalization in Appendix A).

In the segments associated with the torso, for the x and z directions, the largest value (segment) of the normalized geometry is made the 1.0 value. The other segments are given a factor equal to a percentage of that largest value. For the y direction (length), the y values of the normalized figure geometry add up to a total of 100%. Each y value of the normalized segments is divided by that 100% value, and that results in the y factor.

Some segments have two declarations, differentiated by a ".1" or ".2" suffix. These cases denote the use of tapering scaling rather than bounding box scaling. As explained in

[1]Notice that this numeration follows that of Cheverud et al. [19]. The reader is forewarned that in Gordon et al. [23] the numeration has an offset of +1 with respect to the one here presented, i.e., a "one" must be added to each variable number appearing in the formulae.

[2]As with any aspect of the figure model definition, if better "guestimates" are available, it is easy to update the model.

Variable Name	Axis	Formula	Comment
clavicle	x	$(0.5 * [11])$	
upper_arm	x	$((1.0/\pi) * [12])$	$(\pi = 3.14159)$
upper_arm	y	$((1.0/\pi) * [12])$	
upper_arm	z	$([5])$	
lower_arm	x	$((1.0/\pi) * [53])$	
lower_arm	y	$((1.0/\pi) * [53])$	
lower_arm	z	$([88])$	
palm	x	$([58])$	
palm	y	$((2.0/3.0) * (1.0/\pi) * [127])$	((2.0/3.0) guestimate)
palm	z	$(0.6 * [60])$	((0.6) guestimate)
upper_leg	x	$([105])$	
upper_leg	y	$((1.0/\pi) * [104])$	
upper_leg	z	$(-[75] + [108])$	
lower_leg	x	$((1.0/\pi) * [29])$	
lower_leg	y	$((1.0/\pi) * [29])$	
lower_leg	z	$([75] - [76])$	
foot	x	$([76])$	
foot	y	$([51])$	
foot	z	$([10])$	

Figure 5.1: Rules: Intermediate Anthropometric Data Representation

Variable Name	Axis	Formula	Comment
toes	x	$((1.0/3.0) * [76])$	(1.0/3.0 guestimate)
toes	y	$([51])$	
toes	z	$([52] - [10])$	
head	x	$([61])$	
head	y	$([63])$	
head	z	$([255])$	
neck	x	$((1.0/\pi) * [81])$	
neck	y	$((1.0/\pi) * [81])$	
neck	z	$(-[31] + [100] - [255])$	
upper_torso	x	$([33])$	
upper_torso	y	$([37])$	
upper_torso	z	$([31] - [103])$	
center_torso	x	$([113])$	
center_torso	y	$([116])$	
center_torso	z	$(-[68] + [103])$	
lower_torso	x	$([66])$	
lower_torso	y	$([25])$	
lower_torso	z	$([68] - [108])$	
finger	x	$(0.25 * [58])$	
finger	y	$(0.25 * [58])$	
finger	z	$(0.1333 * [60])$	(0.1333 guestimate)

Figure 5.2: Rules: Intermediate Anthropometric Data Representation (continued)

this document, tapering scaling considers both ends of the segment. Case .1 is associated with the proximal end and case .2 is associated with the distal end. Finally, the variables used in the formula of the form *variable – name.axis* refer to the variables and axis of the intermediate representation.

5.2.2 Anthropometric Mensuration Methods

Aside from the figure building capabilities, the figure model provides a set of anthropometric figure mensuration methods. These methods offer the capability of extracting the anthropometric information embedded in a figure model instance. It is important to realize that a figure model instance is, by itself, an entity representing the anthropometric data used to create it. If one places the figure instance in a graphics environment, its anthropometric dimensions (as defined by the figure model) are a representation of the actual anthropometric data within that simulated environment. However, it is necessary for the figure model to provide a way of extracting the measurements that have been embedded in it (and possibly some other extrapolated measurements). The mensuration methods provide this capability.

Primitive mensuration methods have been implemented in Jack. These primitive methods are complemented by (user-) redefinable rules. A list of these rules appears next. These rules implement 69 of the measurements defined in the ANSUR-88 standard.

Mensurations are done in a site to site basis. The definition of the sites, which are established on the (generic) figure definition file, can be as involved as necessary (i.e., consider auxiliary planes, lines, etc.).

The necessary measurement entities (landmarks, sites, planes) need to be defined in the figure model. In the default case, these entities were approximated visually when the generic figure was first defined. It is obvious that the resulting approximation is a very rough. While no default rules are provided by the model to better approximate these entities, the system is prepared to accept such rules as input, were they available. In other words, it is possible to use rules to define the location of any site in the figure model, including mensuration entities (a plane, for example, may be defined using a site, since sites have orientation and position).

As defined in the survey, most of the measurements are not vector distances, but rather distances along a given axis. This is taken into consideration by specifying a measurement axis.

```
// FORMAT :
// P posturename   ---- defines a posture
//
// D variable_name  variable# axis  ---- distance definition
// site1
// site2
//
// C variable_name variable# ---- circumference definition
```

Segname.axis	Formula	Comment
t1.x	$0.5 * upper\_torso.x * 0.8166$	(0.8166 = 81.6% x.max)
t1.y	$(upper\_torso.z + center\_torso.z) * 0.04309$	(0.04309 = 4.3% of 1.0)
t1.z	$0.5 * upper\_torso.y * 0.7150$	(0.7150 = 71.5% z.max)
t2.x	$0.5 * upper\_torso.x * 0.9477$	idem
t2.y	$(upper\_torso.z + center\_torso.z) * 0.04567$	idem
t2.z	$0.5 * upper\_torso.y * 0.8600$	idem
t3.x	$0.5 * upper\_torso.x * 0.9812$	idem
t3.y	$(upper\_torso.z + center\_torso.z) * 0.04235$	idem
t3.z	$0.5 * upper\_torso.y * 0.9178$	idem
t4.x	$0.5 * upper\_torso.x * 1.0000$	(1.0000 = x.max)
t4.y	$(upper\_torso.z + center\_torso.z) * 0.06077$	idem
t4.z	$0.5 * upper\_torso.y * 0.8900$	idem
t5.x	$0.5 * upper\_torso.x * 0.9693$	(0.9603 = 96% x.max)
t5.y	$(upper\_torso.z + center\_torso.z) * 0.04899$	idem
t5.z	$0.5 * upper\_torso.y * 0.9612$	idem
t6.x	$0.5 * upper\_torso.x * 0.9693$	idem
t6.y	$(upper\_torso.z + center\_torso.z) * 0.05193$	idem
t6.z	$0.5 * upper\_torso.y * 0.9677$	idem
t7.x	$0.5 * upper\_torso.x * 0.9693$	idem
t7.y	$(upper\_torso.z + center\_torso.z) * 0.05193$	idem
t7.z	$0.5 * upper\_torso.y * 0.9852$	idem
t8.x	$0.5 * upper\_torso.x * 0.9693$	idem
t8.y	$(upper\_torso.z + center\_torso.z) * 0.05193$	idem
t8.z	$0.5 * upper\_torso.y * 0.9852$	idem
t9.x	$0.5 * upper\_torso.x * 0.9693$	idem
t9.y	$(upper\_torso.z + center\_torso.z) * 0.05488$	idem
t9.z	$0.5 * upper\_torso.y * 0.9900$	(0.9900 = z.max)
t10.x	$0.5 * upper\_torso.x * 0.9693$	idem
t10.y	$(upper\_torso.z + center\_torso.z) * 0.05967$	idem
t10.z	$0.5 * upper\_torso.y * 0.9825$	(0.9825 = 98.2% z.max)
t11.x	$0.5 * upper\_torso.x * 0.9442$	idem
t11.y	$(upper\_torso.z + center\_torso.z) * 0.03867$	idem
t11.z	$0.5 * upper\_torso.y * 0.9600$	idem
t12.x	$0.5 * upper\_torso.x * 0.9442$	idem
t12.y	$(upper\_torso.z + center\_torso.z) * 0.08361$	idem
t12.z	$0.5 * upper\_torso.y * 0.9350$	idem

Figure 5.3: Rules: Target Anthropometric Data Representation

Segname.axis	Formula	Comment
l1.x	$0.5 * upper\_torso.x * 0.9442$	(0.9442 = 95.4% x.max)
l1.y	$(upper\_torso.z + center\_torso.z) * 0.07845$	(0.07845 = 7.8% of 1.0)
l1.z	$0.5 * upper\_torso.y * 0.9000$	(0.9000 = 90% z.max)
l2.x	$0.5 * upper\_torso.x * 0.9275$	idem
l2.y	$(upper\_torso.z + center\_torso.z) * 0.07845$	idem
l2.z	$0.5 * upper\_torso.y * 0.9000$	idem
l3.x	$0.5 * upper\_torso.x * 0.9066$	idem
l3.y	$(upper\_torso.z + center\_torso.z) * 0.05782$	idem
l3.z	$0.5 * upper\_torso.y * 0.9000$	idem
l4.x	$0.5 * upper\_torso.x * 0.9066$	idem
l4.y	$(upper\_torso.z + center\_torso.z) * 0.07514$	idem
l4.z	$0.5 * upper\_torso.y * 0.9000$	idem
l5.x	$0.5 * upper\_torso.x * 0.9066$	idem
l5.y	$(upper\_torso.z + center\_torso.z) * 0.07661$	idem
l5.z	$0.5 * upper\_torso.y * 0.93$	idem

Figure 5.4: Rules: Target Anthropometric Data Representation (continued)

```
// site1
// site2
// site3
// site4

//
// ########  STANDING POSTURE  ########
//
P mystand

D BIDLBDTH 13 0
right_upper_arm.LM_deltoid_pt_rt
left_upper_arm.LM_deltoid_pt_lft

D BIMBDTH 14 0
right_lower_leg.LM_lateral_malleolus
right_lower_leg.LM_medial_malleolus

D BIZBDTH  20 2
bottom_head.LM_right
bottom_head.LM_left

D BUTTDPTH 25 2
lower_torso.LM_buttock_pt_post
lower_torso.LM_front_pelvis
```

Segname.axis[.end]	Formula	Comment
bottom_head.x	$0.5 * (head.y)$	
bottom_head.y	$0.5 * (head.x)$	
bottom_head.z	$(head.z)$	
neck.x	$0.5 * (neck.y)$	
neck.y	$0.5 * (neck.x)$	
neck.z	$(neck.z)$	
upper_arm.1.x	$0.5 * (upper\_arm.y)$	
upper_arm.1.y	$0.5 * (upper\_arm.x)$	
upper_arm.1.z	$(upper\_arm.z)$	
upper_arm.2.x	$0.5 * (lower\_arm.y)$	
upper_arm.2.y	$0.5 * (lower\_arm.x)$	
upper_arm.2.z	1.0	arbitrary
lower_arm.1.x	$0.5 * (lower\_arm.y)$	
lower_arm.1.y	$0.5 * (lower\_arm.x)$	
lower_arm.1.z	$(lower\_arm.z)$	
lower_arm.2.x	$0.5 * (palm.x)$	
lower_arm.2.y	$(palm.z)/3.0$	(3.0 guestimate)
lower_arm.2.z	1.0	arbitrary
upper_leg.x	$0.5 * (upper\_leg.y)$	
upper_leg.y	$0.5 * (upper\_leg.x)$	
upper_leg.z	$(upper\_leg.z)$	
lower_leg.x	$0.5 * (lower\_leg.y)$	
lower_leg.y	$0.5 * (lower\_leg.x)$	
lower_leg.z	$(lower\_leg.z)$	
foot.x	$(foot.z)$	
foot.y	$0.5 * (foot.y)$	
foot.z	$0.5 * (foot.x)$	
toes.x	$(toes.z)$	
toes.y	$0.5 * (toes.y)$	
toes.z	$0.5 * (toes.x)$	
lower_torso.x	$0.5 * (lower\_torso.y)$	
lower_torso.y	$0.5 * (lower\_torso.x)$	
lower_torso.z	$(lower\_torso.z)$	
clavicle.x	$0.5 * (1.0) * 10.0$	1.0, 10.0 arbitrary
clavicle.y	$(1.0) * 10.0$	idem
clavicle.z	$0.5 * (clavicle.x) * 2.0$	2.0 cancels 0.5

Figure 5.5: Rules: Target Anthropometric Data Representation (continued)

Segname.axis[.end]	Formula	Comment
palm.x	$0.5 * (palm.x)$	
palm.y	$(palm.z)$	
palm.z	$0.5 * (palm.y)$	
finger00.1.x	$0.5 * (finger.x)$	
finger00.1.y	$(finger.z)$	
finger00.1.z	$0.5 * (finger.y)$	
finger00.2.x	$0.5 * (finger.x)$	
finger00.2.y	1.0	arbitrary
finger00.2.z	$0.5 * (finger.y)$	
finger10.1.x	$0.5 * (finger.x)$	
finger10.1.y	$(finger.z)$	
finger10.1.z	$0.5 * (finger.y)$	
finger10.2.x	$0.5 * (finger.x)$	
finger10.2.y	1.0	arbitrary
finger10.2.z	$0.5 * (finger.y)$	
finger20.1.x	$0.5 * (finger.x)$	
finger20.1.y	$(finger.z)$	
finger20.1.z	$0.5 * (finger.y)$	
finger20.2.x	$0.5 * (finger.x)$	
finger20.2.y	1.0	arbitrary
finger20.2.z	$0.5 * (finger.y)$	
finger30.1.x	$0.5 * (finger.x)$	
finger30.1.y	$(finger.z)$	
finger30.1.z	$0.5 * (finger.y)$	
finger30.2.x	$0.5 * (finger.x)$	
finger30.2.y	1.0	arbitrary
finger30.2.z	$0.5 * (finger.y)$	

Figure 5.6: Rules: Target Anthropometric Data Representation (continued)

Segname.axis[.end]	Formula	Comment
finger01.1.x	$0.5 * (finger.x)$	
finger01.1.y	$(finger.z)$	
finger01.1.z	$0.5 * (finger.y)$	
finger01.2.x	$0.5 * (finger.x)$	
finger01.2.y	1.0	arbitrary
finger01.2.z	$0.5 * (finger.y)$	
finger11.1.x	$0.5 * (finger.x)$	
finger11.1.y	$(finger.z)$	
finger11.1.z	$0.5 * (finger.y)$	
finger11.2.x	$0.5 * (finger.x)$	
finger11.2.y	1.0	arbitrary
finger11.2.z	$0.5 * (finger.y)$	
finger21.1.x	$0.5 * (finger.x)$	
finger21.1.y	$(finger.z)$	
finger21.1.z	$0.5 * (finger.y)$	
finger21.2.x	$0.5 * (finger.x)$	
finger21.2.y	1.0	arbitrary
finger21.2.z	$0.5 * (finger.y)$	
finger31.1.x	$0.5 * (finger.x)$	
finger31.1.y	$(finger.z)$	
finger31.1.z	$0.5 * (finger.y)$	
finger31.2.x	$0.5 * (finger.x)$	
finger31.2.y	1.0	arbitrary
finger31.2.z	$0.5 * (finger.y)$	

Figure 5.7: Rules: Target Anthropometric Data Representation (continued)

Segname.axis[.end]	Formula	Comment
finger02.x	$0.5 * (finger.x)$	
finger02.y	$(finger.z)$	
finger02.z	$0.5 * (finger.y)$	
finger12.x	$0.5 * (finger.x)$	
finger12.y	$(finger.z)$	
finger12.z	$0.5 * (finger.y)$	
finger22.x	$0.5 * (finger.x)$	
finger22.y	$(finger.z)$	
finger22.z	$0.5 * (finger.y)$	
finger32.x	$0.5 * (finger.x)$	
finger32.y	$(finger.z)$	
finger32.z	$0.5 * (finger.y)$	
thumb0.1.x	$0.5 * (finger.y)$	
thumb0.1.y	$(finger.z)$	
thumb0.1.z	$0.5 * (finger.x)$	
thumb0.2.x	$0.5 * (finger.y)$	
thumb0.2.y	1.0	arbitrary
thumb0.2.z	$0.5 * (finger.x)$	
thumb1.1.x	$0.5 * (finger.y)$	
thumb1.1.y	$(finger.z)$	
thumb1.1.z	$0.5 * (finger.x)$	
thumb1.2.x	$0.5 * (finger.y)$	
thumb1.2.y	1.0	arbitrary
thumb1.2.z	$0.5 * (finger.x)$	
thumb2.x	$0.5 * (finger.y)$	
thumb2.y	$(finger.z)$	
thumb2.z	$0.5 * (finger.x)$	

Figure 5.8: Rules: Target Anthropometric Data Representation (continued)

```
D CHSTBDTH 33 0
t7.LM_chest_right
t7.LM_chest_left

D CHSTDPTH 37 2
t7.LM_chest_back
t7.LM_chest_front

D HEADBRTH 61 0
bottom_head.LM_tragion_rt
bottom_head.LM_tragion_lft

D HIPBRTH 66 0
lower_torso.LM_buttock_pt_rt_lat
lower_torso.LM_buttock_pt_lft_lat

D BISBDTH 15 0
lower_torso.LM_hip_joint_rt
lower_torso.LM_hip_joint_lft

D INPUPBTH 69 0
right_eyeball.LM_pupil_rt
left_eyeball.LM_pupil_lft

D WSTBRTH 113 0
14.LM_waist_rt
14.LM_waist_lft

D WSTDPTH 116 2
14.LM_waist_post
14.LM_waist_ant_navel

D WRTHLGTH 131 2
right_lower_arm.LM_stylion
right_thumb2.LM_thumbtip

D INFORBB 240 2
bottom_head.LM_back
bottom_head.LM_infraorbitale_r

D ACRGHT 3 1
right_foot.LM_bottom_rfoot
right_upper_arm.LM_acromion_r

D CERVHT 31 1
```

```
right_foot.LM_bottom_rfoot
neck.LM_cervicale

D CRCHHGHT 39 1
right_foot.LM_bottom_rfoot
lower_torso.LM_crotch_level_rt

D ILCRSIT 68 1
right_foot.LM_bottom_rfoot
15.LM_iliocristale

D KNEEHTPMP 73 1
right_foot.LM_bottom_rfoot
right_lower_leg.LM_midpatella

D LATFEMEP 75 1
right_foot.LM_bottom_rfoot
right_upper_leg.LM_lat_femoral_epicondyle_standing

D LATMALHT 76 1
right_foot.LM_bottom_rfoot
right_lower_leg.LM_lateral_malleolus

D STATURE 100 1
right_foot.LM_bottom_rfoot
bottom_head.LM_TOP_HEAD

D SUPSTRHT 102 1
right_foot.LM_bottom_rfoot
t1.LM_suprasternale

D TENRIBHT 103 1
right_foot.LM_bottom_rfoot
13.LM_tenth_rib

D TROCHHT 108 1
right_foot.LM_bottom_rfoot
lower_torso.LM_trochanterion_rt

D WSTHNI 119 1
right_foot.LM_bottom_rfoot
14.LM_waist_rt

D ECTORBT 233 1
bottom_head.LM_ectoorbitale_rt
bottom_head.LM_TOP_HEAD
```

```
D TRAGT 255 1
bottom_head.LM_tragion_rt
bottom_head.LM_TOP_HEAD

D ACRDLG 5 3
right_upper_arm.LM_acromion_r
right_upper_arm.LM_radiale

D BLFTLG 10 3
right_foot.LM_pternion
right_foot.LM_first_metatarsophalangeal_protrusion

D BCRMBDTH 11 3
right_upper_arm.LM_acromion_r
left_upper_arm.LM_acromion_l

D FTBRHOR 51 3
right_foot.LM_first_metatarsophalangeal_protrusion
right_foot.LM_fifth_metatarsophalangeal_protrusion

D FOOTLG 52 3
right_foot.LM_pternion
right_toes.LM_acropodion

D HANDBRTH 58 3
right_finger00.LM_metacarpale_II
right_finger30.LM_metacarpale_V

D HANDLG 60 3
right_lower_arm.LM_stylion
right_finger12.LM_dactylion_III_r

D HEADLGTH 63 3
bottom_head.LM_glabella
bottom_head.LM_back

D INSCYE1 70 3
right_clavicle.LM_midscye_rt
left_clavicle.LM_midscye_lft

D RASTL 88 3
right_upper_arm.LM_radiale
right_lower_arm.LM_stylion

D SHOULGTH 93 3
```

```
right_upper_arm.LM_acromion_r
neck.LM_trapezius_pt_rt

D WRINFINGL 130 3
right_lower_arm.LM_stylion
right_finger02.LM_dactylion_II


// --------- Circumferences -----------
C  ANKLCIRC 6
right_lower_leg.LM_ankle_back
right_lower_leg.LM_ankle_front
right_lower_leg.LM_ankle_left
right_lower_leg.LM_ankle_right

C  CALFCIRC 29
right_lower_leg.LM_back
right_lower_leg.LM_front
right_lower_leg.LM_left
right_lower_leg.LM_right

C  KNEECIRC 72
right_lower_leg.LM_midpatella
right_lower_leg.LM_midp_back
right_lower_leg.LM_midp_left
 right_lower_leg.LM_midp_right

C  NECKCIRC 81
neck.LM_back
neck.LM_front
neck.LM_neck_rt_lat
neck.LM_neck_lft_lat

C  THGHCIRC 104
right_upper_leg.LM_back
right_upper_leg.LM_front
right_upper_leg.LM_left
right_upper_leg.LM_right

C  WRISCIRC 127
right_lower_arm.LM_styl_back
right_lower_arm.LM_styl_front
right_lower_arm.LM_styl_left
right_lower_arm.LM_styl_right

//
```

```
// ########  SITTING POSTURE  ########
//
P mysit

D BUTTKLTH 27 2
lower_torso.LM_buttock_pt_post
right_lower_leg.LM_knee_pt_ant

D BUTTPLTH 28 2
lower_torso.LM_buttock_pt_post
right_upper_leg.LM_popliteal_sit

D ACRHTST 4 1
lower_torso.LM_sit_level
right_upper_arm.LM_acromion_r

D CERVSIT 32 1
lower_torso.LM_sit_level
neck.LM_cervicale

D EYEHTSIT 50 1
lower_torso.LM_sit_level
bottom_head.LM_ectocanthus

D KNEEHTSIT 74 1
right_foot.LM_bottom_rfoot
right_lower_leg.LM_suprapatella

D POPHGHT 87 1
right_foot.LM_bottom_rfoot
right_upper_leg.LM_popliteal_sit

D SITTHGHT 94 1
lower_torso.LM_sit_level
bottom_head.LM_TOP_HEAD

D THGHCLR 105 1
lower_torso.LM_sit_level
right_upper_leg.LM_thigh_front

D WSTHSTNI 121 1
lower_torso.LM_sit_level
14.LM_waist_rt

D WSHTSTOM 122 1
lower_torso.LM_sit_level
```

```
l4.LM_waist_ant_navel


//
// ########  FUNCTIONAL POSTURE  ########
//
P functionalzero

D  THMBTPR 107 2
t4.LM_back_pt_rt
right_thumb2.LM_thumbtip

D WRWALLLN 132 2
t4.LM_back_pt_rt
right_lower_arm.LM_stylion

P functionalone

D WRWALLEX 133 2
t4.LM_back_pt_rt
right_lower_arm.LM_stylion

//
// ########  SITTING FOREARM-UP  ########
//
P sitting_forearm_up
D lower_torso.LM_sit_level right_lower_arm.LM_olecranon_flex 1 49 ELRHGHT

//
// ########  STAND FOREARM-UP    ########
//
P forearm_up

D FORHDLG  55 2
right_lower_arm.LM_olecranon_flex
 right_finger12.LM_dactylion_III_r

D SHOUELLT  92 3
right_upper_arm.LM_acromion_r
right_lower_arm.LM_olecranon_flex

//
// ########  FOREARM-UP-HAND-CLOSED  ######
//
P forearm_up_hand_closed
```

```
D  WRCTRGRL  126  2
right_lower_arm.LM_stylion
right_palm.LM_hand_closed_center

//
// ########  BICEPS FLEX  #######
//
P biceps-flex
// --------- Circumferences -----------
C  BICIRCFL 12
right_upper_arm.LM_back
right_upper_arm.LM_front
right_upper_arm.LM_left
right_upper_arm.LM_right

C FCIRCFL 53
right_lower_arm.LM_back
right_lower_arm.LM_front
right_lower_arm.LM_left
right_lower_arm.LM_right

//
// ########## OVERHEAD POSTURE ########
//
P overhead
D OVHDFTRH  84 1
right_foot.LM_bottom_rfoot
right_finger12.LM_dactylion_III_r

//
// ########## ARM SPAN POSTURE ########
//
P arm_span
D SPAN  99 0
right_finger12.LM_dactylion_III_r
left_finger12.LM_dactylion_III_l
```

5.2.3 Error in Measurements

Circumferences are estimated from the following equation.

$$circumf(d_1, d_2) = \pi * \sqrt{(\frac{d_1^2 + d_2^2}{2})} \tag{5.1}$$

where d_1, d_2 are the major and minor axis (or vice versa) in the measurement's

cross-section. Notice this is an approximation for "circumference" anthropometric measurements.

As for linear measurements, the following aspects introduce error:

- Location of Measurement Sites:
  While the measurement sites have been carefully defined, it must be understood that the geometry of the body does not exhibit a one-to-one correspondence to a real human being. The shape of the torso, for instance, is as simple as possible to reduce the polygonal complexity of the figure. Definition of measurement sites must take this into account.

- The type of geometry used to model the body:
  Rigid (i.e., non deformable) are employed, and therefore, sites like "lower_torso.LM_sit_level" can only be defined approximately to account for any deformations.

- Measurements that span several segments can show inaccuracies. Assuming a normalization technique is employed for scaling, where the distance between joint centers is normalized, the only linear measurements that are guaranteed are link lengths, as defined in Appendix E. Lengths other than link lengths, involving more the one link, cannot be guaranteed. There are various reasons for this:

  - The mechanical joint model used in the body does not necessarily correspond to a real human joint. For instance, the shoulder complex is modeled with two mechanical joints, that up to a certain degree, simulate the real case. For instance, the accuracy of a functional reach, which involves the shoulder complex, is limited by the accuracy of the shoulder complex motion itself.
  - Measurements spanning several links usually consider soft tissue (body fat, skin), e.g., buttock-popliteal length. Soft tissue is not modeled by the Polybody, except for the geometrical characteristics of the surfaces themselves. The scaling process used for the sizing of the polyhedral surfaces does not consider overall shape, but only link length and (certain) circumferences. Beyond that, the enfleshment of the polybody is that provided by the generic (artistically modeled) surfaces.
  - Somatotype modeling is not considered. To understand what this implies, think about scaling the body using an average person's measurements. If the shape of the geometry used to model the body is fixed (at design time) then problems may be encountered when trying to model an obese (or a thin) person with the same geometry used to model an average size person.

5.3 Joints

Each joint in the human body has a *range of motion.* The range of motion of a joint, described in terms of angles, is measured in degrees for each degree of freedom (DOF), that is, for each plane in which movement is allowed at a joint.

The human figure model of *Jack* allows motion at sixty–eight (68) joints which have a total of one–hundred–thirty–five (135) DOF. For each DOF two measurements are required, an upper limit and a lower limit, that is, two–hundred– seventy (270) joint measurements for each human figure. The default data for the joints is extracted from [38], [30] and [17].

The following list shows the corresponding degrees of freedom of each joint. The associated data (not shown) is extracted from [38] (or [17]). Spine joint limits are extracted from [30]. Other values not appearing in this list have been "guestimated".

```
NECK (U. Limit)
x       neck lateral right
y       neck flexion
z       neck rotation right
NECK (L. Limit)
x       neck lateral left
y       neck extension
z       neck rotation left
SHOULDER (U. Limit)
x       shoulder, abduction
y       shoulder, flexion
z       shoulder, rotation lat
SHOULDER (L. Limit)
x       shoulder, adduction (from O.B. by D. Chaffin)
y       shoulder, extension
z       shoulder, rotation med
ELBOW (U. Limit)
y       elbow, flexion
ELBOW (L. Limit)
y       elbow extension
WRIST (U. Limit)
x       wrist radial
y       wrist flexion
z       forearm pronation
WRIST (L. Limit)
x      wrist ulnar
y      wrist extension
z      forearm supination
HIP (U. Limit)
x      hip abduction
y      hip flexion
z      hip lateral rotation prone (O.B. by D. Chaffin)
HIP (L. Limit)
x      hip adduction (O.B. by D. Chaffin)
y      hip extension
z      hip medial rotation prone (O.B. by D. Chaffin)
```

Joint Name	*Site-A*	*Site-B*	*Type*
left_eyeball	bottom_head.left_eyeball	left_eyeball.base	xz
right_eyeball	bottom_head.right_eyeball	right_eyeball.base	xz
atlanto_occipital	neck.distal	bottom_head.proximal	zyx
base_of_neck	t1.distal	neck.proximal	yzx
solar_plexus	t1.distal	upper_torso.proximal	NONE
left_clavicle_joint	upper_torso.lclav	left_clavicle.proximal	xy
right_clavicle_joint	upper_torso.rclav	right_clavicle.proximal	-xy
left_shoulder	left_clavicle.lateral	left_upper_arm.proximal	zxy
right_shoulder	right_clavicle.lateral	right_upper_arm.proximal	-z-xy
left_elbow	left_upper_arm.distal	left_lower_arm.proximal	y
right_elbow	right_upper_arm.distal	right_lower_arm.proximal	y
spinet2t1	t2.distal	t1.proximal	xyz
spinet3t2	t3.distal	t2.proximal	xyz
spinet4t3	t4.distal	t3.proximal	xyz
spinet5t4	t5.distal	t4.proximal	xyz
spinet6t5	t6.distal	t5.proximal	xyz
spinet7t6	t7.distal	t6.proximal	xyz
spinet8t7	t8.distal	t7.proximal	xyz
spinet9t8	t9.distal	t8.proximal	xyz
spinet10t9	t10.distal	t9.proximal	xyz
spinet11t10	t11.distal	t10.proximal	xyz
spinet12t11	t12.distal	t11.proximal	xyz
spinel1t12	l1.distal	t12.proximal	xyz
spinel2l1	l2.distal	l1.proximal	xyz
spinel3l2	l3.distal	l2.proximal	xyz
spinel4l3	l4.distal	l3.proximal	xyz
spinel5l4	l5.distal	l4.proximal	xyz
waist	lower_torso.distal	l5.proximal	yzx
left_hip	lower_torso.lhip_lateral	left_upper_leg.proximal	zxy
right_hip	lower_torso.rhip_lateral	right_upper_leg.proximal	-z-xy
left_knee	left_upper_leg.distal	left_lower_leg.proximal	-y
right_knee	right_upper_leg.distal	right_lower_leg.proximal	-y
left_ankle	left_lower_leg.distal	left_foot.proximal	zxy
right_ankle	right_lower_leg.distal	right_foot.proximal	-z-xy
left_toes	left_foot.toes	left_toes.proximal	y
right_toes	right_foot.toes	right_toes.proximal	y

Figure 5.9: Joint Description (hands in separate figure)

Joint Name	*Site-A*	*Site-B*	*Type*
left_wrist	left_lower_arm.distal	left_palm.base	yxz
right_wrist	right_lower_arm.distal	right_palm.base	y-x-z
linfinger02	left_finger01.tip	left_finger02.base0	-x
linfinger01	left_finger00.tip	left_finger01.base0	-x
lmidfinger12	left_finger11.tip	left_finger12.base0	- x
lmidfinger11	left_finger10.tip	left_finger11.base0	- x
lringfinger22	left_finger21.tip	left_finger22.base0	-x
lringfinger21	left_finger20.tip	left_finger21.base0	-x
lpinfinger31	left_finger30.tip	left_finger31.base0	- x
lpinfinger32	left_finger31.tip	left_finger32.base0	- x
left_finger00	left_palm.f11	left_finger00.base0	z-x
left_finger10	left_palm.f22	left_finger10.base0	z-x
left_finger20	left_palm.f33	left_finger20.base0	z-x
left_finger30	left_palm.f44	left_finger30.base0	z-x
lthumb2	left_thumb1.tip	left_thumb2.base0	-x
lthumb1	left_thumb0.tip	left_thumb1.base0	-x
lthumb0	left_palm.thumb0	left_thumb0.base0	-zy
rinfinger02	right_finger01.tip	right_finger02.base0	x
rinfinger01	right_finger00.tip	right_finger01.base0	x
rmidfinger12	right_finger11.tip	right_finger12.base0	x
rmidfinger11	right_finger10.tip	right_finger11.base0	x
rringfinger22	right_finger21.tip	right_finger22.base0	x
rringfinger21	right_finger20.tip	right_finger21.base0	x
rpinfinger31	right_finger30.tip	right_finger31.base0	x
rpinfinger32	right_finger31.tip	right_finger32.base0	x
right_finger00	right_palm.f11	right_finger00.base0	zx
right_finger10	right_palm.f22	right_finger10.base0	zx
right_finger20	right_palm.f33	right_finger20.base0	zx
right_finger30	right_palm.f44	right_finger30.base0	zx
rthumb2	right_thumb1.tip	right_thumb2.base0	x
rthumb1	right_thumb0.tip	right_thumb1.base0	x
rthumb0	right_palm.thumb0	right_thumb0.base0	-z-y

Figure 5.10: Joint Description (hands)

```
KNEE (U. Limit)
y     knee flexion
KNEE (L. Limit)
y     knee extension
ANKLE (U. Limit)
y     ankle, dorsi
ANKLE (L. Limit)
y     ankle, plantar
KNUCKLES (U. Limit)
y     flexion
KNUCKLES (L. Limit)
y     extension
```

5.4 Mass

The global mass is the sum of the masses of all body segments. In the Jack human model, the mass of each segment is computed, according to [3], as a percentage of the total mass, *gblmass*. The percentages are average percentile values for a fit male population as found in the NASA male crewmember trainees. For the average general population or a population which is skewed to either the small/light weight or large/heavy weight these percentages will vary. The formula used to compute each segment's mass appears to the right of the segment's name. These formulae are (user-) redefinable.

```
bottom_head          0.0790*gblmass
left_eyeball         0.0001*gblmass
right_eyeball        0.0001*gblmass
neck                 0.0020*gblmass
left_clavicle        0.0300*gblmass
right_clavicle       0.0300*gblmass
t1                   0.3128*gblmass/17
t2                   0.3128*gblmass/17
t3                   0.3128*gblmass/17
t4                   0.3128*gblmass/17
t5                   0.3128*gblmass/17
t6                   0.3128*gblmass/17
t7                   0.3128*gblmass/17
t8                   0.3128*gblmass/17
t9                   0.3128*gblmass/17
t10                  0.3128*gblmass/17
t11                  0.3128*gblmass/17
t12                  0.3128*gblmass/17
l1                   0.3128*gblmass/17
l2                   0.3128*gblmass/17
l3                   0.3128*gblmass/17
l4                   0.3128*gblmass/17
```

```
15                  0.3128*gblmass/17
left_upper_arm      0.0280*gblmass
right_upper_arm     0.0280*gblmass
left_lower_arm      0.0160*gblmass
right_lower_arm     0.0160*gblmass
left_palm           0.0040*gblmass
right_palm          0.0040*gblmass
left_finger32       0.0020*gblmass/15
left_finger31       0.0020*gblmass/15
left_finger30       0.0020*gblmass/15
left_finger22       0.0020*gblmass/15
left_finger21       0.0020*gblmass/15
left_finger20       0.0020*gblmass/15
left_finger12       0.0020*gblmass/15
left_finger11       0.0020*gblmass/15
left_finger10       0.0020*gblmass/15
left_finger02       0.0020*gblmass/15
left_finger01       0.0020*gblmass/15
left_finger00       0.0020*gblmass/15
left_thumb2         0.0020*gblmass/15
left_thumb1         0.0020*gblmass/15
left_thumb0         0.0020*gblmass/15
right_thumb2        0.0020*gblmass/15
right_thumb1        0.0020*gblmass/15
right_thumb0        0.0020*gblmass/15
right_finger32      0.0020*gblmass/15
right_finger31      0.0020*gblmass/15
right_finger30      0.0020*gblmass/15
right_finger22      0.0020*gblmass/15
right_finger21      0.0020*gblmass/15
right_finger20      0.0020*gblmass/15
right_finger12      0.0020*gblmass/15
right_finger11      0.0020*gblmass/15
right_finger10      0.0020*gblmass/15
right_finger02      0.0020*gblmass/15
right_finger01      0.0020*gblmass/15
right_finger00      0.0020*gblmass/15
lower_torso         0.1260*gblmass
left_upper_leg      0.1000*gblmass
right_upper_leg     0.1000*gblmass
left_lower_leg      0.0460*gblmass
left_lower_leg      0.0460*gblmass
right_lower_leg     0.0460*gblmass
left_foot           0.0126*gblmass
right_foot          0.0126*gblmass
left_toes           0.0014*gblmass
```

Segment	X-axis	Y-axis	Z-axis
Head	0.5 * Bitragion Breadth	Tragion-Wall Depth	0.17 * Tragion Vertex Ht
Up. Arm	assume Symmetry	assume Symmetry	0.48 * Link Length
Lo. Arm	assume Symmetry	assume Symmetry	0.41 * Link Length
Hand	Symmetry at z-axis loc.	Symmetry at z-axis loc.	0.51 * palm length
Up. Leg	assume Symmetry	assume Symmetry	0.41 * Link Length
Lo. Leg	assume Symmetry	assume Symmetry	0.44 * Link Length
Foot	Symmetry at z-axis loc.	Symmetry at z-axis loc.	0.44 * foot length (from heel)

Figure 5.11: Segment Center of Mass Computation

`right_toes` `0.0014*gblmass`

5.5 Center of Mass Computation

The center of mass of the segments in the figure is computed based on [3], and [38], as shown in Fig. 5.11, and/or the segment's surface geometry.

5.6 Special Body Constructs

There are four "special" body constructs:

- Shoulder Complex. For detailed explanation of the shoulder complex refer to [6] and [51].
- Torso-Spine. The torso-spine model is considered in detail in [35] and [6].
- Hand. The hand model is presented in the next section of this document.
- Foot "Complex". The foot is modeled with two (rigid) segments: base of foot (also known as ball_of_foot), defined from the heel to the ball of the foot, and the toes segment, defined from the ball of the foot to the toe tips. The toes segment shows no toe detail (neither segments nor joints). For the foot "complex", two joints are considered, namely, ankle and toes. The latter connects the base of the foot to the toes segment. The former connects the base of the foot to the lower leg segment.

5.7 Anthropometry of the Hand

The human model includes a detailed model of the hands. The modeling of the hand has been done in accordance to [2], [14], and [31]. The fingers are modeled with three segments each, and three joints. The Phalanges include Proximal, Middle, and Distal segments. For the Thumb, the First Metacarpal segment is included so that it has three segments as the other digits. The joints of the fingers are as follows:

- the Trapeziometacarpal joint between the Metacarpal segment of the thumb and the palm, with two degrees of freedom (extension-flexion and adduction- abduction),
- the Metacarpophalangeal joint between the palm and the phalangeal proximal segment of the digit, with two degrees of freedom (extension-flexion and adduction-abduction), except thumb with one degree of freedom (extension-flexion),
- the Interphalangeal joint between the proximal and middle, and the middle and distal phalanges, with one degree of freedom each (extension-flexion).

The palm geometry has been shortened to consider the proper connection sites with the digits (from the skeletal point of view). This results in a hand which appears to have "long" fingers and "short" palm, i.e., closer to the skeletal counterpart, and biomechanically more correct.

Note: Even though this approach gives (in general) good results, several points must be made clear:

- In a real human hand, the proximal Phalange of Digit 2 (index finger's proximal phalange) starts at a location closer to the wrist than the proximal phalanges of Digits 3, 4, and 5.
- The palm model used in the human figure assumes, that all Digits 2, 3, 4, and 5 all start at the same location (collinear).
- The overall result of making this assumption is that Digit 2 will sometimes appear larger than Digit 3, which in general is incorrect.

The palm consists of one segment, with 6 joints, 1 for each digit and 1 for the wrist, i.e., no modeling of carpals or metacarpals (the carpal segment of the thumb is modelled, but is considered part of the thumb and not the palm) is done.

5.7.1 Scaling

The hand scaling is based on [24]. Figure 5.12 describes the mapping between the anthropometric data and the geometry of the hand. For circumference measurements divide by π to obtain the required dimension.

5.7.2 Hand Joint Limits

The joint limits of the fingers are obtained from: [9], [14], [20], and [29], and can be seen in figure 5.13.

These references consider range of motion rather than joint limits, i.e., the data comes as a total range. This total range has been broken in upper and lower limits based on educated guesses and empirical observations. In the future, if more accuracy is required then the necessary data must be provided.

Segment/Digit	x/y	z
Finger 0 (digit2)	-	10
0	13,14	21
1	13,14	20
2	15,16	19
Finger 1 (digit3)	-	22
0	25,26	33
1	25,26	32
2	27,28	31
Finger 2 (digit4)	-	34
0	37,38	45
1	37,38	44
2	39,40	43
Finger 3 (digit5)	-	46
0	49,50	57
1	49,50	56
2	51,52	55
Thumb (digit1)	-	01
0	4,5	07
1	4,5	08
2	4,5	09

Figure 5.12: Anthropometric Survey Data Mapping

Joint	Range of Motion (degrees)
Finger DIP	Ext-Flx(60)
Finger PIP	Ext-Flx(100)
Finger MCP	Ext-Flx(90) Abd-Add(60)
Thumb IP	Ext-Flx(85)
Thumb MCP	Ext-Flx(50)
Thumb CMC	Ext-Flx(50) Abd-Add(40)

Figure 5.13: Joint Limits

5.8 Female

Currently there is no geometry for the female human body model (i.e., no female Polybody). To overcome this situation, the male body can be used instead, scaled according to the female anthropometric values, and replacing the male head with a head with female features for improved appearance.

Obviously, this is not the best solution as it introduces problems due to the difference in proportionality, and segment alignment between the sexes, e.g., hip breadth, breasts, etc. A "native" female Polybody will be available in the future.

Notice that the default mass factors provided are for a male population (see Section 5.4.).

Chapter 6

Statistical Methods

6.1 Introduction

In this chapter, the statistical methods for population data manipulation are presented.

For the statistical manipulation of anthropometric data, three methods are available: Monte Carlo family simulation, Cadre family simulation, and multinormal conditional estimation of proportionality patterns.

6.2 Proportionality Reconstruction

The notion of proportionality can be easily understood if one thinks of the (body) proportions of one particular individual. For the population case, the concept of proportionality preservation cannot be easily compared with its individual-based counterpart. Proportionality refers to a particular subject, and, yet, one has a population of individuals. Ultimately, a given proportionality will always be identified with a single subject. However, for the population case, we will use what we refer to as "statistical" subjects (also known as synthetic subjects). A statistical subject is an artificial entity that represents a given aspect of the population. Furthermore, rather than considering individual statistical subjects, one is interested in families of such subjects, as explained in the next section.

From a statistics standpoint, proportionality preservation translates to observance of the correlations between the variables of interest, when applying a given statistical model. In our case, we consider proportionality preservation from two fronts. First, we are interested in preservation of proportions, when using population sample data to scale a figure model. Second, we want to be able to reconstruct (part of) the data necessary for the scaling of a figure model, starting from incomplete (partial) sets of anthropometric data. The first front, preservations of proportions, will be considered when figure families are discussed. We proceed to discuss the second front, proportionality reconstruction next.

6.2.1 Reconstruction Method

In the following discussion we assume the reader has a basic knowledge of the multinormal distribution model. For further details refer to Morrison [36].

Assuming that the covariance matrix of the multinormal distribution is full rank, i.e., the distribution is nonsingular, the following properties hold:

- Given an N-dimensional random vector $\bar{X}$, with a multinormal distribution, mean $\bar{\mu}$ and covariance matrix Σ of rank N. For any $M * N$ real matrix Q, with rank $M \leq N$, the vector:

$$\bar{A} = Q\bar{X} \tag{6.1}$$

  is a multinormal random variable. Furthermore, its mean is $E(\bar{A}) = Q\bar{\mu}$ and its covariance matrix is $Q\Sigma Q^T$.

- Given a $(p+q)$ -dimensional multinormal population, with variates represented by the random vector $\bar{X} = [\bar{X}_0, \bar{X}_1]$, where $\bar{X}_0$ is of size p and $\bar{X}_1$ is of size q. The mean vector is:

$$\bar{\mu} = [\bar{\mu_0}, \bar{\mu_1}]^T \tag{6.2}$$

  and the covariance matrix is:

$$\Sigma = \begin{bmatrix} \Sigma_{00} & \Sigma_{01} \\ \Sigma_{10} & \Sigma_{11} \end{bmatrix} \tag{6.3}$$

  where Σ_{00} has dimensions $p * p$, Σ_{01} has dimensions $p * q$ and Σ_{11} has dimensions $q * q$. Also, $\Sigma_{10} = \Sigma_{01}^T$.

  $\bar{X}_0$ and $\bar{X}_1$ are multinormal random vectors. The distributions of these vectors are $N(\bar{\mu_0}, \Sigma_{00})$ and $N(\bar{\mu_1}, \Sigma_{11})$, respectively. Σ_{01}'s values, relative to those of Σ_{00} and Σ_{11}, determine the degree and pattern dependence between the two sets of variates.

- A random vector $\bar{X}$, (partitioned into two random sub-vectors $\bar{X}_0$ and $\bar{X}_1$), with mutivariate normal distributions $\bar{\mu}$ and Σ, will be independently distributed if and only if $\Sigma_{01} = 0$.

- The conditional density function of $\bar{X}_0$, assuming elements of $\bar{X}_1$ are fixed to $\bar{x_1}$, is expressed as:

$$g(\bar{x_0}|\bar{x_1}) = \frac{f(\bar{x_0}, \bar{x_1})}{h(\bar{x_1})} \tag{6.4}$$

  where f represents the joint density function of the complete set of $p+q$ variates and h is the joint density function of the q fixed variates.

- It can be proven [36] that the conditional distribution of a set of p variates from a multinormal population, with q other variates of the population held fixed, is multinormal with (conditional) mean vector:

$$\bar{\mu}_c = \bar{\mu}_0 + \Sigma_{01}\Sigma_{11}^{-1}(\bar{x}_1 - \bar{\mu}_1) \tag{6.5}$$

and (conditional) covariance matrix:

$$\Sigma_c = \Sigma_{00} - \Sigma_{01}\Sigma_{11}^{-1}\Sigma_{01}^{T} \tag{6.6}$$

The implications of the above results are as follows:

- Given a p-dimensional multinormal random population with mean $\bar{\mu}$ and covariance matrix Σ, and given a random vector $\bar{X}_1$ of size q, extracted from that population, such that $q \leq p$, it is possible to estimate a random vector $\bar{X}_0$, with dimension $p - q$, based on the given (fixed) q values, thus obtaining a (complete) p-dimensional model, with $\bar{X}_0$ having mean $\bar{\mu}_c$ and covariance matrix Σ_c and $\bar{X}_1$ having mean $\bar{\mu}$ and covariance matrix Σ.
- Therefore, if we have *a priori* knowledge of the correlations of a p-dimensional multinormal random population and extract q (key) scaling features out of an individual from that population, we can estimate the remaining $p - q$ values.

6.3 Families of Figures

In workspace design problems, it is common to find a need to consider multivariate situations, in which the correlations among all of the variables needs to be taken into consideration. The traditional (but unfortunately incorrect) practice of considering (univariate) percentile-based data in the construction (scaling) of figure models leads to undesirable results such as the trio of all 5th-, all 50th-, and all 95th-percentile dummies.

The idea of a population accommodation test is that one must verify all the individuals of a given population (or if Monte Carlo simulation is used, all the elements of a sample with the appropriate size – the size is determined such that the simulation converges to a solution) to find out if each of them fits the given accommodation problem (the accommodation percentage is the fraction of all the individuals tested that actually fit the environment). The problem with this approach is that the number of individuals that must be tested is equal to the population (sample) size (or a significantly large size for Monte Carlo simulations).

Another approach referred to as the boundary cases considers only "extreme" points along the hypersurface of the multinormal distribution of the given population. Such approach is significant not only because it reduces considerably the number of individuals needed to be tested, but, most importantly, because it focuses attention on "important" cases. In practice, the Monte Carlo and Cadre families complement each other.

In the next section, we review both the Monte Carlo and the population "extremes" approach. However, before we proceed in that direction, let us briefly explain first the notion of a "statistical individual".

Statistical individuals are an abstraction that consolidates a number of anthropometric variables across a given population. A population (statistical) individual is produced as a result of a statistical transformation of the population data. When one performs statistical transformations of anthropometric data, the individuality of each observation (population subject) disappears, to lead towards general (common) traits that define the population. It is necessary to use proportionality preserving statistical transformations so that correlations among variables are considered.

6.3.1 Monte Carlo Simulation

The multivariate correlation model can be applied in simulations of body proportionality and space requirements. These simulations fall under the type denominated "Monte Carlo" [33].

Controlled randomness is the key to these simulations. The generation of human figure model anthropometric parameters is turned into a stochastic process. A given number of anthropometric dimension patterns is randomly generated, hoping that one particular instance fits a set of desired design constraints.

Algorithmic randomness simulates real world randomness. This randomness is controlled using correlations obtained from real world population data, thus making the stochastic process approximate the "randomness" of the real world.

When a random vector of correlated variables is generated to simulate, say, the dimensional parameters of the human body, a set of (probabilistic) requirements is being satisfied, and yet one can claim that only with a certain probability, in N experiments, a generated pattern satisfies the constraints under consideration; it could be the first try or the last one, or the constraint could be satisfied in all of them, or in none of them. Monte Carlo simulations have been applied to the problem of space accommodation [10, 11]. In this context, N experiments are carried out and tested against a given set of spatial requirements. Statistics are obtained from the results of this simulation, to find out what percentage of the population is accommodated within the space under analysis.

If one has access to the population raw data, one could try the following alternative analysis. Pick, randomly, samples of N individuals out of the population, and test their measurements against the given space constraints. Then, obtain statistics of the results to figure out the size of the accommodated population. One can determine appropriate sampling sizes (i.e., sizes of N), that can provide, with a given probability, satisfactory results.

Roebuck states:

> "The result of [a Monte Carlo simulation] is [$\cdots$] a large set of synthetic operators (typically 400-3600), each having dimensions of varying percentiles within the manikin design and each dimension statistically distributed with

essentially the same means and standard deviations and with all pairs of dimensions correlated to the same degree of the original population. So, although no single synthetic operator is guaranteed to match a living subject, each operator represents one possible case that could occur in a population of living people, without violating any of the underlying statistics" [28].

6.3.1.1 Implementing a Monte Carlo family

We start by assuming that the covariance matrix Σ is symmetric positive definite [15]. Since every square matrix Q is similar to a matrix J in Jordan canonical form, and if M is a modal matrix for Q, then, in particular, for the square covariance matrix Σ, it holds that

$$\Sigma = MJM^{-1} \tag{6.7}$$

J is a diagonal matrix of the eigenvalues of Σ. M is composed of the eigenvectors of Σ. Therefore, $MM^{-1} = \mathbf{I}$. In fact, being the covariance matrix symmetric positive definite, $M^{-1} = M^T$.

Let us define a random variable

$$\bar{A} = M^T \bar{X} \tag{6.8}$$

where $\bar{A}$ is a multinormal random variable, with mean $E(\bar{A}) = M^T \bar{\mu}$ and covariance matrix $M^T \Sigma M$. Since $\Sigma = MJM^T$, then it follows that the covariance matrix of $\bar{A}$ is J.

We are fabricating a (whitening) transformation that yields a random vector with covariance equal to the identity matrix (i.e., a random vector whose variates are not only independent but have a unitary standard deviation).

Observing that the square root of a positive definite matrix Q is given by

$$Q^{1/2} = MJ^{1/2}M^{-1} \tag{6.9}$$

and noticing that, if Q is diagonal, then

$$Q^{1/2} = J^{1/2}. \tag{6.10}$$

Finally, let us define

$$\bar{B} = J^{-1/2} \bar{A} \tag{6.11}$$

where $\bar{B}$ is a multinormal random variable with mean $E(\bar{B}) = J^{1/2}M^T\bar{\mu}$ and covariance matrix $J^{-1/2}M^T\Sigma M(J^{-1/2})^T = J^{-1/2}M^TMJ^{-1}M^TM(J^{-1/2})^T = \mathbf{I}$.

Now, suppose that we generate a standardized (multinormal) random vector $\bar{\delta}$ [13, 37], with mean $\bar{\mu} = 0$ and covariance matrix $\Sigma = \mathbf{I}$. Then, by means of the inverse of the transformation just defined, we can transform that random vector into another one with a specific mean and covariance. It can be proven easily that given a random vector $\bar{\delta}$, with mean $\bar{\mu} = 0$ and covariance matrix $\Sigma = \mathbf{I}$, and applying the transformation

$$\bar{X} = MJ^{1/2}\bar{\delta} \tag{6.12}$$

we obtain a multinormal random vector $\bar{X}$ with covariance matrix $\Sigma = MJM^T$. Furthermore, if we add the mean vector $\bar{\mu}$ to $\bar{X}$, we obtain a multinormal random vector with mean $\bar{\mu}$.

Each such vector represents a member of the Monte Carlo family.

6.3.2 The Cadre Family

The Cadre[1] family (refer to Bittner et al. [12]), also known as the Boundary family, is an alternative approach to the Monte Carlo approach. Bittner demonstrates that, in general, for accommodation testing purposes, the Cadre family is at least as good as its Monte Carlo counterpart. The Cadre family exhibits features such as having the ability to screen the multivariate space in a systematic fashion, and the ability to substantially reduce the size of testing space, while capturing a significant amount of variance. This goal is accomplished by systematically considering only extreme cases of the population.

Roebuck states the following regarding boundary manikins:

> "These boundary conditions are much better able to guarantee a given percentage of accommodation in design than are common-percentile manikins", [28, p. 95] .

It is important to understand several aspects regarding the Cadre family:

- A Cadre family is a multivariate representation of the extremes of the population distribution. For example, given n variables, the Cadre family consists of $2^n + 2n + 1$ subjects. The problem is that, as n becomes big (> 7), we observe a combinatorial explosion of extreme cases. To "delay" the occurrence of such explosion of combinations, instead of using the variables directly in the application of the method, we make use of the principal factors with larger associated variance, capturing a significant percentage of the total variation.
- The Cadre family is a "statistical" representation of the extremes of a population. This implies that none of the Cadre members may actually have a real-world

[1] Cadre: From the French for frame, limits, scope.

counterpart. This is an issue that should be carefully analyzed when building mock-ups of the Cadre family simulation.

- Each of the family members represents an orthogonal trait. A trait can be understood as a linear combination of (some of) the original variables. This characteristic may be useful when using the family for accommodation design purposes. However, making sense of the traits in terms of the original variables is, in general, not an easy undertaking.

6.3.2.1 Implementing a Cadre family

Our implementation considers Bittner et al. [12] and Meindl et al. [32]. For details about principal components and the multivariate normal distribution, refer to Morrison [36].

The probability density function of the multivariate normal distribution is defined by:

$$f(\bar{x}) = ((2\pi)^p|\Sigma|)^{-\frac{1}{2}} exp(-\frac{1}{2}(\bar{x}-\bar{\mu})^T\Sigma^{-1}(\bar{x}-\bar{\mu})) \tag{6.13}$$

where p is the number of dimensions, $\bar{x}$ is a random vector, μ and Σ are the mean and variance-covariance matrix of the population, respectively.

Any symmetric matrix (e.g., Σ) has an associated quadratic form, $Q(\bar{x})$. The quadratic form is defined by

$$Q(\bar{x}) = \bar{x}^T\Sigma\bar{x} \tag{6.14}$$

By equating this quadratic form with a constant value, k, the resulting expression defines a conic surface. The shape of the conic surface depends on the diagonal elements of Σ. For Σ representing a variance-covariance matrix, the conic surface described is a hyper-ellipsoid.

Vector $\bar{x}$ defines specific locations of the surface of the hyper-ellipsoid. For the case of $p = 2$, the case of an ellipse, vector $\bar{x}$ is collinear with the gradient vector of $Q(\bar{x})$ at four directions. These directions are those of the principal axes of the ellipse. In general, for any given p, vector $\bar{x}$ is collinear with the gradient vector of $Q(\bar{x})$ at $2 * p$ directions.

The key points about the principal components are:

- The computation of the principal components is done such that each of the components has a correlation of 0 with the other components.
- The variances associated with the principal components are not all equal to 1 (notice the original variables have variance equal to one). The sum of the variance of the principal components will be p, the number of variables. However, the components are computed such that the first component captures the largest variance, the second one captures the next largest variance, and so forth. In consequence, a few of components capture most of the variance. This fact can be used to justify the canceling of components that do not contribute significantly to the total variance.

The generation of the Cadre family involves the following steps.

- First, the value of constant k, defining the hyper-ellipsoid size, must be determined, for a given accommodation percentage. This basically translates to finding the value of the constant k of the quadratic form defined above. Meindl et al. suggest finding this value by iteration.

- The next step in the solution is to find the component scores. This is done by computing the principal components of the covariance matrix.

  The principal components, P_i $(i = 1, \cdots, k)$, are linear combinations of the original variables. P_1 has the largest sample variance, P_2 has the next largest variance, and so on:

$$P_i = a_{1,1}Z_1 + a_{2,1}Z_2 + \cdots + a_{p,1}Z_p \tag{6.15}$$

  for p variables Z_1,Z_2,$\cdots$, Z_p. The computation of the weight coefficients $a_{i,j}$ is done considering the following constraint:

$$\sum_{k=1}^{p} a_{k,j}^2 = 1. \tag{6.16}$$

  The components are prioritized according to their associated captured variance (descending order), and as many are kept as for the sum of the associated variances to add up to a desired total captured variance.

- In the next step, the component scores matrix is multiplied by a binary matrix formed by all combinations of $\pm k$. Also, one may want to compute the axial points. This can be done by forming a matrix defining, in the same row, a single $+k$ and a $-k$ in two adjacent columns, with each row having exactly two elements different than 0. This step yields standard score vectors.

- The final step is to convert the standard scores into raw scores.

The family generated should include not only the hyperellipsoid surface points, but also the axial points and the mean. The total number of subjects produced is $2^m + 2m + 1$, where m is the number of factor loadings as determined by the minimum amount of variance specified.

Our implementation supports input as raw data, or correlations and means and standard deviations, or covariances and means. By establishing a series of parameters such as accommodation percentage and percentage of variance sought, a given number of Cadre subjects is automatically determined.

Chapter 7

Generator of Figures

7.1 Genfig

The *Genfig* (**Gen**erator of **Fig**ures) program marks the start of a new way of thinking about human figure modeling. The program departs from the style of its predecessor, Spreadsheet System for Anthropometric Scaling (SASS), towards a more user-friendly, anthropometric and multivariate statistics based framework.

Following the spirit of *SASS*, *Genfig* offers a combination of capabilities that allow the user to construct anthropometrically based human figure models. *Genfig* introduces new ideas in anthropometric modeling. *Genfig* is not an updated version of *SASS*. *Genfig* has been designed and implemented as an entirely new program. In fact, *Genfig* is only a shell program that presents a front end to a set of standalone modules. In this way, new modules can be easily added into *Genfig*, using a "plug & play" approach. For example, the figure constructor module, currently designed for handling the construction of Polybody figure instances, could be easily replaced by another module designed for, say, handling the construction of Smooth-Skin figure instances.

Currently, there are 4 "plug & play" modules in *Genfig*.

- **Figure Constructor**: Handles the creation of *Peabody* grammar based figure definition files, following the rules presented in Chapter 5.
- **Solver**: Implements the statistical conditional reconstruction process presented in the previous chapter.
- **Monte Carlo Family Generator**: Implements the production of Monte Carlo families of figures, as explained in the previous chapter.
- **Cadre Family Generator**: Implements the production of a Cadre family, as explained in the preceding chapter.

Each of these modules is a standalone program by itself, that can be run from the command line. *Genfig* has no knowledge about the internal implementation of these

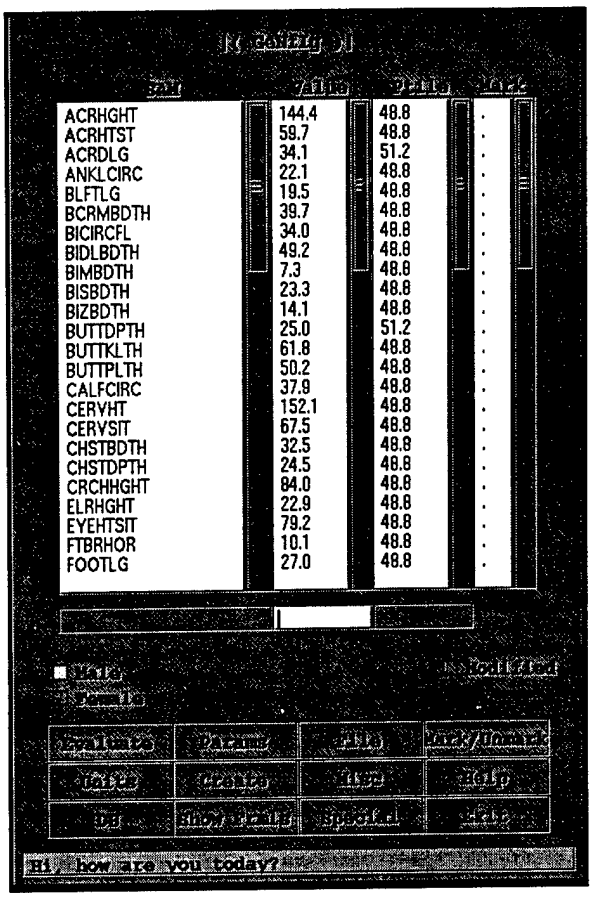

Figure 7.1: Graphical User Interface of Genfig

modules. All it knows is what the interface (parameter list) of each program exposes. In this way, it is easy to see how the implementation of any of these modules can be modified, as long as the interface remains intact.

In the future, the user will be allowed to add new modules into the functionality of *Genfig*, or maybe acquire them from a third-party vendor. Currently, however, it is only possible to replace any of the existing modules by comparable ones (same interface). For example, a new constructor module considering Smooth-Skin figures could replace the default one.

Genfig is implemented as a standalone module complementing the Jack software package. Fig. 7.1 shows the X-windows based user interface of the program.

Additional information on *Genfig* may be found in its user manual (to be published as part of the "Jack's User Manual"). Also, refer to the previous chapter of this document for detailed information on the statistical methods. The next two sections present a brief overview of the two main methods associated with figure modeling, i.e., the constructor and the measurer. The former is one of the modules in *Genfig*. The latter is implemented as a module in *Jack* .

7.2 The constructor

The rules used by the constructor method have been explained in Chapter 5 and will not be repeated here.

The constructor module takes anthropometric data (see the list of default variables in Appendix refa6).

All the variable declaration, data associated with those variables, and methods to handle that data, which are required by the constructor to build figure model instances, are expected as input to the program. The module provides defaults for all these inputs, stored in plain text files. The module has been designed to work in an interpreted fashion. Each time a figure instance is created, the program interprets the methods (rules) and data needed for the job. The reader concerned by the efficiency of the interpretation process should not be. In fact, the constructor has been optimized as to be about 30 times faster than the figure generator in *SASS*. Those concerned about the added difficulty of writing scripts for the program should not be either. While, originally, we considered the option of using a traditional interpreted language, such as LISP, for the scripting, we ended up deciding on a much simpler, but powerful enough, script format, based exclusively on variable declarations and arithmetic expressions.

The output of the constructor module is a figure definition file (an example is presented in Appendix C).

7.3 The Measurer

In addition to the *Genfig* program, the anthropometric measurement capabilities in *Jack* have been changed to support the functionality of *Genfig*. The measurement methods of the figure model are implemented in *Jack* .

The measurements are done all at once, through the execution of the *Jack* command measure all. As a result of this command, a file with the name of the figure being measured, but with the extension *.dmp*, is created. This file contains the value for each of the variables defined in the measurement rules. The list of the default measurement rules has been presented in Chapter 5. As with all other aspects of *Genfig*, the set of default measurement rules is fully user-redefinable. In this way, the rules can be updated or replaced easily.

Chapter 8

Anthropometric Error

8.1 Anthropometric Error Model

In this chapter, a brief overview of anthropometric error is presented. For a more in depth discussion of this topic, refer to [5].

8.1.1 Anthropometric Error

To address the problem of quantifying the accuracy of the approximations used in the application of anthropometric ($1D$) data to the construction of a $3D$ figure model, the notion of anthropometric error, AE, is introduced.

We define anthropometric error as follows:

> "Anthropometric error is the difference between the value of an anthropometric measurement extracted out of a human figure instance, referred to as set D_2, and the corresponding anthropometric measurement value used for the scaling of that figure instance, i.e., $AE = D_2 - D_1$."

8.1.2 Causes of AE

Anthropometric error in a human figure can have three main causes:

- **The scaling data:**
  For example, the $1D$ nature of anthropometric measurements imposes limitations on usability of such data in the construction of anthropometric figures. Another example is the use of univariate percentile data, rather than multivariate data, in the construction of the figure instances.
- **The geometry and link structure:**
  As explained earlier in this document, anthropometric human models are limited by the degree of control offered by the geometry (shape and sizing). Limited control leads to approximations, which themselves lead to error.

- **The approximation equations used to match data and geometry/link-structure:**
  See, for example, the approximation equations used in building the Polybody figure in Chapter 5. These equations consider only a small subset of all anthropometric measurements available in the ANSUR-88 survey. As their name indicates, the evaluation of these equations results in approximate values, each of which corresponds to a specific anthropometric parameter used in the construction of the figure instances. The use of approximations leads to error.

  Notice that in the case of *Genfig*, it is very easy to modify, enhance, or even replace the default set of equations with a set of better equations. The developing of higher quality approximation equations requires further research, and future efforts should address this problem.

8.1.3 *AE* Model

By itself, the *AE* associated with the different anthropometric variables is a passive entity, namely, a collection of error values.

An *AE* model is an abstraction that considers *AE* not only from the standpoint of a collection of values, but also introduces data measurement and analysis methods. The implementation of the *AE* model results in a tool designed for testing the anthropometric error of both specific individual figures or of entire families of figures.

As a result of such tests, an $m * n$ anthropometric error matrix, E is produced, where m is the number of variables tested and n is the size of the testing family (an individual can be considered a degenerate case of a family of one member). The error matrix, E, presents the error for each of the variables and each of the figures involved in the testing.

The *AE* measurement model tool provides some basic data analysis methods. However, the user is expected to have access to, say, a spreadsheet tool or, even better, a statistical analysis package for more in depth analysis of the data, if deemed necessary. Some of those tools follows:

- Conversion of E into $|E|$. This function allows the user to convert all entries of E into positive values. A negative entry in E means that the measured value on the figure model is smaller than the control value. A positive entry means the measured value is bigger than the control value. This information is important, but in certain occasions, what is needed is not to know the direction of the error but its magnitude. Ideally, the matrix E should be $\mathbf{0}$. Matrix $|E|$ allows us to better see how far E is from the ideal without concern for signs. For example, if we compute the mean of a given row of E, the negative values and the positive values may cancel and therefore result in a mean close to 0, which would give us the incorrect impression that the population *AE* for that particular row is close to ideal. Using $|E|$, on the other hand, results in a better understanding of the actual size of the mean *AE* for a given row.

- Data filtering. This function detects which rows (variables) are affected by the different figures in the population. It yields E_R, a reduced version of E that contains

only the rows (variables) which are affected by the different figures. For example, if the value of variable X is constant across all figures in the family being tested, then, the row for X is not included in E_R. This allows the user to concentrate only on the variables that are affected by the figure type.

- Means, standard deviations and correlations. These are basic statistics and correlations that summarize the data in E. The correlations can tell us how the $AE's$ are correlated with each other. After all, the $AE's$ themselves follow a multivariate distribution.

The AE modeling tool makes use of the Monte Carlo and Cadre families presented earlier in this document. The use of these families in the framework of AE is somewhat different to their use in workspace design. In the AE framework, the use of these families of figures permits AE-based testing of the scalability of a given figure model, to all the possible combinations found in a given anthropometric family. The (data) members of these families are scaling patterns (vectors), one per member. Each (data) member is tested by using it in the scaling of the given figure model, and, then, the $AE's$ of the resulting figure instance are measured. While the members of the Monte Carlo family represent random cases of the corresponding multinormal distribution, the Cadre family members represent the extreme cases of such distribution. Using the Cadre family for AE-based testing of the scalability of the figure model allow us to determine the anthropometric quality of the members of the resulting figure family, each of which represents a worst-case scaling condition.

Chapter 9

Inverse Dynamics

9.1 Introduction

The inverse dynamics code in *Jack* has been rewritten, in an effort to expand its capabilities and add flexibility. The new interface allows the user to pick an arbitrary part of any articulated figure (including but not limited to the standard human figure), and compute forces at the joints for both static and dynamic postures. Force information can now be obtained for any part of the articulated human spine, arms, and legs. Simulated external forces (such as extra weight) can be applied anywhere on the figure to analyze their effect on the joint forces. A brief description of how the updated inverse dynamics system works follows next.

9.2 Introduction to Inverse Dynamics

Inverse dynamics is the calculation of the forces required at a human figure's joints in order to produce a given set of joint accelerations. The term "force" is used to refer to both translational forces and rotational forces (torques). Static figure postures are a special case where the joint accelerations are all zero. However, even in the static case, forces at the joints exist to counteract the external forces acting on the figure (such as the gravitational force).

For the inverse dynamics calculations, all the external forces acting on a figure should be known. One does not need to worry about the weight of the segments, which is automatically computed by *Jack* . There are other forces though, which are, in general, unknown or hard to compute. The most common such example is the contact forces between figure segments and the environment. Fortunately, there is a way to get around this problem in the most common cases without much trouble: a root site can be defined for the part of the figure of interest. One can think of the root as the point where the figure is attached to the world. The rest of the figure segments are hinged in a tree structure from the root through joints. If there are no other contacts between the part of the figure we are interested in and the environment, then nothing else needs to be done. For example, if one is studying the forces at the joints in the upper human body, as long as no other upper

body segment is attached to the environment, the root can be set at the pelvis. There is no problem if there are external forces, all of which known, as the root can be placed in any segment.

The placement of the root defines an internal tree hierarchy in the figure. Each segment is connected to its parent via a joint. Inverse dynamics computes the torques at the joints as well as the force acting on a segment from the segment's parent. The magnitude of this force is particularly important in reference to evaluating the stress at the lower human back, given a particular posture.

The most efficient way available to calculate inverse dynamics is using the recursive Newton-Euler dynamic equations. The particular implementation in *Jack* is based on the work of Roy Featherstone [1].

9.3 Inverse Dynamics in *Jack*

Inverse dynamics in *Jack* allow the user to compute static and dynamic joint forces for stationary and moving postures, respectively. The user can select the part of the body that is of interest, assign the root to any site on the figure and apply external forces on any figure segment. Motions can be created using the standard *Jack* motion generation commands. The joint velocities and accelerations, as well as the translational velocity and acceleration of the root site, are used to compute the dynamic joint forces.

Torques at the joint are computed around the defined joint axes. Forces acting on a segment are displayed both in the local segment frame (segment coordinates), as well as in the inertial frame (world coordinates). The units used throughout the calculations are [Kg, m, s].

All the available commands can be executed from the force submenu of *Jack* or can be typed in directly just like any other JCL command.

9.4 Commands

This section describes the commands used to access the inverse dynamics force data. They are found in the force submenu of *Jack* :

create force data - This command is used to set up the inverse dynamics system. Knowing which joints are of interest in a particular joint force calculation, the user defines the tree structure that contains all of them by selecting the root site (described above) and any number of leaf segments. Only segments contained in the paths from the root to the leaves will be part of force calculation. As an example, to measure joint forces for the right hand of a human figure, one might select `right_clavicle_lateral` as a starting site and `right_palm` as an ending segment. Alternatively, if one is interested in all upper body forces, one might choose `lower_torso_proximal` for the start site and `right_palm` and `left_palm` for the end segments. To execute the command,

[1] Roy Featherstone, Robot Dynamics Algorithms, Kluwer Academic Publishers, 1987.

the user has to press Esc after the last segment is entered. All the segments which become part of the force calculations are displayed in the Log window. Multiple parts of the figure can be marked for inverse dynamics calculations by repeatedly executing this command.

delete force data - This command resets all previously defined inverse dynamics tree structures for a figure.

display static joint torque - This command prints out the static joint torques at a particular joint. Static joint torques refer to the torques needed to maintain a posture when the figure has zero velocity and acceleration (the only forces acting are the weight and any external force the user defined on the figure).

display static segment force - This command displays the three-dimensional force that a segment is receiving from it's parent in the tree hierarchy. For example to find out the force on the lowest segment of the human spine one might ask for the force on the fifth lumbar segment, 15. The force is displayed for convenience both in the local coordinate system of the segment and in the inertial (world) coordinate system.

display dynamic joint torque and display dynamic segment force - In order to analyze the joint forces when a figure is in motion one has to follow these steps:

- Use create force data to define the part of the figure that is of interest.
- Create any motions on the figure using the standard *Jack* motion commands.
- Type Go to execute the motion.
- Use the animation window slider or goto time to select a particular figure posture.
- Run display dynamic joint torque or display dynamic segment force to get the joint forces at the particular time instant.

The velocities and accelerations of the root site and the joints are used to calculate the joint forces. Velocities and accelerations are computed under the assumption that there are 30 *Jack* frames per actual time second. One can adjust the frame rate through the set frame rate command.

attach force to site - This command sets a simulated external force on a segment. The force is applied through a particular segment site, and its direction should be given in the world coordinate system. For example, to attach an additional weight w to a site one should enter $(0, w, 0)$ (Note that the weight of an object is equal to it's mass in Kg multiplied by the gravitational acceleration $9.81 m/s^2$).

clear segment forces - This command clears all external forces from a segment (except from the segment's weight of course).

9.5 Reading Frame Data

Determining the dynamic joint forces relies on getting the figure and joint velocities and acceleration from the standard *Jack* channels. The most straightforward way to fill the

channels with data is to use the standard *Jack* motion commands. If however the motion is inserted from precomputed frames through the read frames command then information about the root site channel should also be read. To obtain the correct structure of a frame file, one should use create force data, then generate a simple motion and write the channel data into a file using write frames. The root site channel data will be written in the frame file as well.

Chapter 10

Smooth Skin

The current human model used in the center is based on rigid segments. Such rigid body model provides the ability for real-time manipulation due to its computational simplicity. However, a rigid body model is inadequate to provide a visually satisfying image, mainly because the connection between segments often creates discontinuities. To correct this unwanted property, we designed a control system to deform the geometry.

The subsequent development of computer graphics has provided different tools in modeling deformable material. In this work, "free-form deformation" [47, 40] was chosen to control the human geometry deformation.

10.1 Design Goal

When dealing with deformable human body, people have a misconception that the deformation has to be volume conservative. However, due to the non-homogeneous property of human tissue and the complex structure of human body, the volume conservation property does not hold in human body. For example, when a body builder flexes his muscle, the change of the shape and size of his muscle is different from the muscle change of a regular person. Also, when a person is breathing, the total volume of the body must be different between inhaling and exhaling. This kind of muscle behavior and volume change are case dependent and beyond the scope of this work. Instead, the design goals are focused on the smooth geometry transition between segments around the joint area and the real time performance.

10.1.1 Smooth Geometry Transition of a Joint:

To model a deformable body, one has to decide how the deformation is contributed. Since every joint in human body connects two segments, the deformation is distributed between these two segments. As mentioned above, this work does not deal with material properties. It is reasonable to assume that each segment contributes half of the deformation. That is, if the rotation angle of the joint is θ, the joint cross section is rotated at an angle of $\theta/2$.

10.1.2 Real Time Performance:

One way to control the geometry deformation is to treat the whole body as a contiguous segment. And the change of human posture is totally controlled by manipulating the control mesh. This method guaranties the smooth transition between joints. However, this method is very expansive, since it applies the deformation algorithm on every node whenever a joint is moved.

To solve this problem, we perform the geometry segmentation to minimize the need of application of the deformation algorithm. With the geometry segmentation, when a joint is moved, the deformation algorithm only applies to those segments in which the deformation actually occurred. Other segments simply perform the rigid body motion.

10.2 Geometry Segmentation

The segmentation of the current rigid body human model is based on the human skeleton structure. With this type of segmentation, the current human model can best simulate the human skeleton motion. However, for the purpose of simulating the geometry deformation, the geometry segmentation is based more on the muscle structure than on the skeleton structure.

In human anatomy, a human skeleton is divided into the axial skeleton and the appendicular skeleton [48]. The muscle structure of these two divisions are different. And the deformation is also different.

The axial skeleton consists of pelvis, torso, neck, and head. In the rigid body model, each vertebrate is considered as one unique segment. However, the muscles in this division usually cross over several of these vertebrate segments. When these muscle groups contract (or relax), all these vertebrate segments are deformed. The result of the deformation should form a continuous surface which describes the shape of these muscle groups. To ensure the smoothness between these segments, we use one contiguous segment to simulate the whole axial division.

The appendicular skeleton contains the bones of free appendages, which consists of upper extremities (arms), lower extremities (legs), and girdles which connect to axial division. In contrast to the muscles in the axial division, the muscles in this division usually only cross one joint, and the deformation caused by these muscles is usually limited to the area around the joint. The deformation in this division is much easier to simulate than the deformation in the axial division. Therefore, we segment this division the same way as the articulatory segmentation.

10.3 Algorithm for Free Form Deformation

The first decision to implement free form deformation is to choose the embeded algorithm and the knot vector. There are several algorithms which are commonly used when implementing free form deformation. They are Bezier, piecewise Bezier, B-spline and

NURB based system [21, 45]. The Bezier algorithm is the simplest algorithm to implement. However, this algorithm connects every control node with every segment node. When the control mesh size is increased, the computation becomes too expansive. On the other hand, the NURB system offers the most flexible control of the free form deformation. However, some of the control options that NURB system offered is redundant in this implementation. In our system, we implement closed-end cubic B-spline based free form deformation. The knots in this implementation is equally spaced.

10.4 Control

Control is the most important issue in implementing the deformable model. In this implementation, we adopt the layer construction concept [16] to build the control mechanism. On the higher level, user changes the human posture by adjusting the joints between virtual skeleton segments. When the displacements of these joints are changed, the behavior level control mechanism moves the free form deformation control nodes based on the control algorithm and the new displacements. After that, the segment geometry is decided according to the new locations of the control nodes and the embedded free form deformation algorithm.

The behavioral control level, which is hidden from user, is more complex. It controls the actual deformation of the geometry based on the joint angles. Since the muscle structures are different, axial division and appendicular division should be considered differently.

10.4.1 Axial Division

There are two types of joints in the axial division. The joints connecting spine segments are dependent joints. Due to the muscle structure, these joints always move as a group. The joints connecting clavicle segments can move independently from the spine joints. However, since the muscle groups which control the movement of the clavicle segments connect to all twelve thoracic vertebrae, the geometry of the clavicle segments are also influenced by the spine joints.

In the implementation, we assigned two control meshes on the axial division. The lower mesh controls the lower torso in which the geometry is changed only when the spine joints are changed. The upper control mesh controls the upper part of the human torso. In this region, both spine movement and shoulder movement will affect the geometry deformation.

10.4.2 Appendicular Division

In appendicular skeleton structure, there are various types of joints. Ball joint, saddle joint, hinge joint, and pivot joint [48], to name a few. These joints are either one degree of freedom joints or three degrees of freedom joints. Different types of joints create different types of skeleton movement. However, with the muscle groups wrapping around the skeleton system, the geometry deformation from different types of joint are not significantly different from one another. We classify these effects into two types, bending and twisting. The

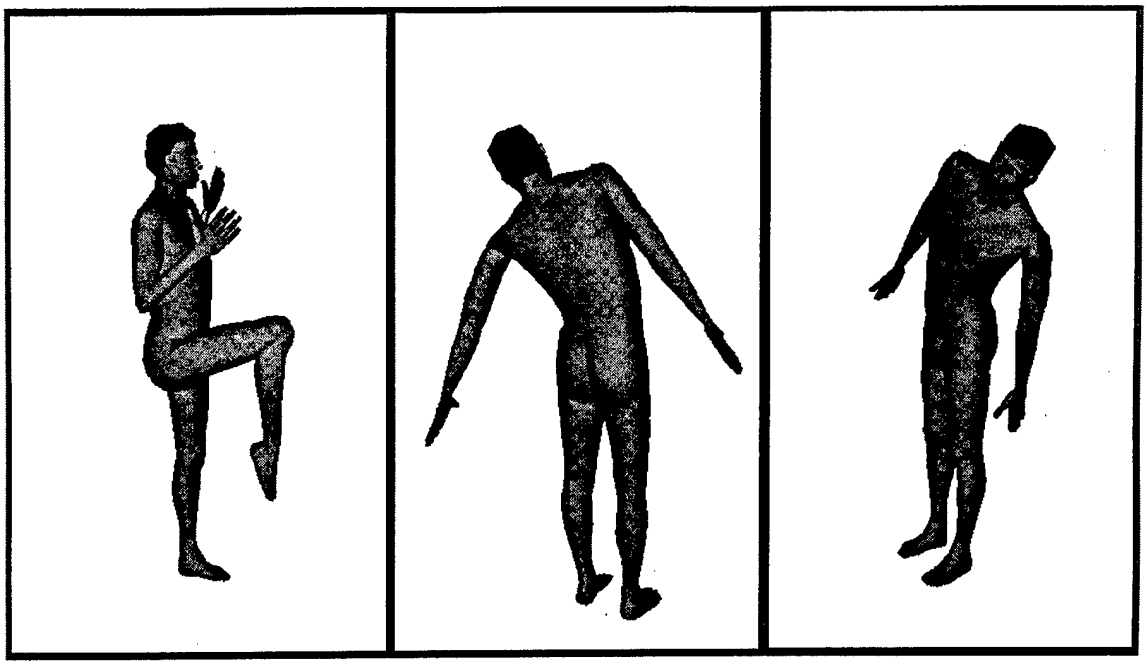

Figure 10.1: Deformable human model

bending effect occured when the rotation axis is perpendicular to the principle axis of the geometry. The twisting effect occured when the rotation axis is parallel to the principle axis of the geometry. These two different types of effect deform different parts of the geometry. In implementation, we assigned three deformation meshes onto each appendicular segment. Two end meshes which are located on top of the joint area, are used to simulate the bending effect. Another mesh which is located in the center of the geometry is used to control the twisting effect.

10.5 Result

Fig. 10.1 is the result of this implementation. The figure on the left demonstrates the deformation of the appendicular division. The figure on the center and the right figures are bending examples of the axial division.

Chapter 11

Estimation of Landmark Locations

As was stated in Chapter 1, there are no $3D$ landmark locations measured in a traditional anthropometric survey. Educated guesses are necessary to estimate these locations. Following are some of the criteria used to place landmarks on the current model. All landmarks on the model are user redefinable.

1. lower_torso
    1. *LM_abdominal_pt* : There is no abdomen of the figure, so the center point on the most protruding face of the lower stomach of the sitting figure is assumed to be abdominal point
    2. *LM_buttock_pt* : point along the middle face of the leg at the level of the buttock point
    3. *LM_gluteal_furrow_pt2* : since there is no distinct thigh in the figure, assume that the gluteal furrow point is the center point where the lower face of the buttock comes into the leg
2. upper_arm
    1. *LM_acromion* : There is no direct line from the neck. Used the skeleton to match to the highest point on the tip of the shoulders
    2. *LM_biceps_pt* : since the figure does not flex, assume that the biceps point is 20
3. toes
    1. *LM_acropodion*: Since there are no toes on the figure, the point farthest away from the heel is assumed to be the acropodion
4. clavicle

1. *LM_axillary_fold_post*: there is no axillary posterior fold, so it is estimated to be directly under the clavicle
2. *LM_clavicle_pt*: the point on the figure that corresponds with the clavicle point on the skeleton

5. neck
   1. *LM_cervicale*: the corresponding edge on the neck of the figure with the seventh cervical vertebra

6. bottom_head
   1. *LM_chin*: most protruding point in the center of the chin
   2. *LM_ear_bottom*: the lowest protruding point on the earlobe
   3. *LM_ear_top*: the highest protruding point on the ear
   4. *LM_ectocanthus*: there are no eyelids on the figure, so the outer corner where the top and bottom of the eye come together is assumed to be the ectocanthus
   5. *LM_ectoorbitale*: the point on the figure that corresponds with the ectoorbitale on the skeleton
   6. *LM_gonion*: point figure that corresponds with the point on the skeleton

7. finger
   1. *LM_dactylion_II*: the outermost tip of the right index finger
   2. *LM_dactylion_III*: the outermost tip of the middle finger

8. upper_leg
   1. *LM_dorsal_juncture*: the center point where the two planes from the calf and the thigh intersect

9. lower_arm
   1. *LM_elbow_crease*: when the elbow is flexed, the center point where the two faces of the forearm and the upper arm intersect

10. foot
    1. *LM_fifth_metatarsophalangel_protrusion*: the point on the skeleton that corresponds with the figure
    2. *LM_first_metatarsophalangeal_protrusion*: the point on the skeleton that corresponds with the figure

Appendix A

Normalization and Scaling of Body Segments

The following is the general algorithm used to prepare the geometry of a given segment for scaling, i.e., segment normalization:

- Given a body segment, find the two cross sections of the segment geometry through the proximal and distal joints (sites) in the x-y plane perpendicular to the z (major) axis of the segment. Compute the x, y bounding rectangle of each the proximal and the distal cross sections: *distal_x_min*, *distal_x_max*, *distal_y_min*, *distal_y_max*, and similarly for the proximal.
- Using the bounding rectangle dimensions, find the scale factors *distal_scale_x*, *distal_scale_y*, *proximal_scale_x*, *proximal_scale_y* in the x and y directions, which map the rectangles to $-1 < x < 1$ and $-1 < y < 1$. Where the scale factor is infinite set it to 1.0 by definition.
- For the z axis, the segment length (distance between proximal and distal joint) must be used. The proximal joint maps to $z = 0$ and the distal joint maps to $z = 1$. The representation of the segment could extend outside the range $[0, 1]$ due to possible overlaps.
- The segment has now its proximal and distal cross sections in a standard position. Notice that these transformations are associated with the geometry of the segment, and have nothing to do with the anthropometric data thus far.

After normalization, we can apply the anthropometric data to scale the segment, as follows:

- Suppose the following data is available (either from a population or from a given individual): segment length, segment proximal circumference, segment distal circumference. Convert the circumferences to x, y scale factors at the appropriate joint. This can be done by computing the major and minor axes of the ellipse with

the arclength equal to the circumference in question, so that the ratio of major to minor (or vice versa, depending on which segment dimension x or y is bigger) axis length is the same as the *distal_scale_x* to *distal_scale_y* (*proximal_scale_x* to *proximal_scale_y*) ratio at each joint. A simpler option is to assume both x, y are equal.

- Transform the segment geometry according to the length scale factor, that is, $new\_segment\_length = segment\_length(scale) * normalizedlength$
- Transform the circumference of the segment by a scale transformation which varies along the z axis of the segment from 0 to *new_segment_length*, that is, $local\_segment\_scale\_x = ((new\_segment\_distal\_scale\_x) - (new\_segment\_proximal\_scale\_x)) * z + new\_segment\_proximal\_scale\_x$. This is a linear scaling in the x direction. The process is similar for the y direction.

Appendix B

Normalization Sites - Link System

The following table shows the sites used during normalization of each of the segments. If no sites were involved, it means the geometry was normalized based on a bounding box approach (i.e., $max - min$). The use of normalization sites assures that the distance between the two sites is a "true" distance. This is necessary to consider the overlapping that occurs between (certain) segments. This distance, is the one that is effectively scaled when the scaling value is applied. Notice that, the sites are only two, usually along the longest axis of the segment. For the other two scaling axes, the approach used is always a bounding box approach.

The distance between the two joint center sites represents the link length associated to a given segment. For the torso segments, the link length (i.e., torso length) is spread across the 17 segments.

Segment Name	Site 1	Site 2
foot	NS_proximal	NS_distal
toes	*bounding box*	*used*
lower_leg	NS_proximal	NS_distal
upper_leg	NS_proximal	NS_distal
bottom_head	NS_proximal	NS_distal
neck	NS_proximal	NS_distal
clavicle	*bounding box*	*used*
upper_arm	NS_proximal	NS_distal
lower_arm	NS_proximal	NS_distal
t1	NS_proximal	NS_distal
t2	NS_proximal	NS_distal
t3	NS_proximal	NS_distal
t4	NS_proximal	NS_distal
t5	NS_proximal	NS_distal
t6	NS_proximal	NS_distal
t7	NS_proximal	NS_distal
t8	NS_proximal	NS_distal
t9	NS_proximal	NS_distal
t10	NS_proximal	NS_distal
t11	NS_proximal	NS_distal
t12	NS_proximal	NS_distal

Figure B.1: Normalization Sites

l1	NS_proximal	NS_distal
l2	NS_proximal	NS_distal
l3	NS_proximal	NS_distal
l4	NS_proximal	NS_distal
l5	NS_proximal	NS_distal
lower_torso	NS_proximal	NS_distal
palm	NS_proximal	NS_distal
finger32	NS_proximal	NS_distal
finger31	NS_proximal	NS_distal
finger30	NS_proximal	NS_distal
finger22	NS_proximal	NS_distal
finger21	NS_proximal	NS_distal
finger20	NS_proximal	NS_distal
finger12	NS_proximal	NS_distal
finger11	NS_proximal	NS_distal
finger10	NS_proximal	NS_distal
finger02	NS_proximal	NS_distal
finger01	NS_proximal	NS_distal
finger00	NS_proximal	NS_distal
thumb2	NS_proximal	NS_distal
thumb1	NS_proximal	NS_distal
thumb0	NS_proximal	NS_distal

Figure B.2: Normalization Sites (continued)

Appendix C

The Figure File

The following is an example of the figure files created by **Genfig v.6.0.** The files are structured as follows:

- Declaration and definition of attributes.
- Declaration and definition of segments.
- Declaration and definition of joints.
- Declaration and definition of strength equations.
- Declaration of posture files.
- Declaration of root site.

```
figure {
    attribute eyewhite {
        rgb = (1,0.89,0.89);
        ambient = 0.19;
        diffuse = 0.76;
    }
    attribute navy {
        rgb = (1,1,1);
        ambient = 0.00;
        diffuse = 0.01;
    }
    attribute eyes {
        rgb = (1,0.49,0.37);
        ambient = 0.09;
        diffuse = 0.34;
    }
    attribute shoes {
        rgb = (1,1,1);
```

```
        ambient = 0.04;
        diffuse = 0.16;
    }
    attribute pants {
        rgb = (0.20,0.20,1);
        ambient = 0.10;
        diffuse = 0.90;
    }
    attribute flesh {
        rgb = (1,0.68,0.57);
        ambient = 0.20;
        diffuse = 0.80;
    }
    attribute upper_lip {
        rgb = (1,0.47,0.47);
        ambient = 0.15;
        diffuse = 0.60;
    }
    attribute hair {
        rgb = (1,0.55,0.08);
        ambient = 0.07;
        diffuse = 0.27;
    }
    attribute lower_lip {
        rgb = (1,0.52,0.52);
        ambient = 0.17;
        diffuse = 0.70;
    }
    attribute shirt {
        rgb = (1,1,1);
        ambient = 0.20;
        diffuse = 0.80;
    }
    archive = "human-5.9.a";
    segment left_eyeball {
        psurf = "EyeBall66.pss" * scale(2.22,0.73,4.68);
        attributes = (eyewhite,navy,eyes);
        centerofmass = (0.00,0.00,2.34);
        mass = 7.85g;
        site base->location = trans(0.00cm,-1.66cm,2.34cm);
        site sight->location = xyz(90.00deg,0.00deg,90.00deg) *
trans(0.00cm,1.01cm,2.34cm);
        site iris->location = trans(0.00cm,-1.66cm,-1.41cm);
        site point->location = trans(0.00cm,-1.66cm,2.45cm);
        site point1->location = xyz(-180.00deg,0.00deg,-180.00deg) *
trans(0.00cm,-1.66cm,3.45cm);
```

```
        site LM_pupil_lft->location = trans(0.00cm,1.01cm,2.34cm);
    }
    segment right_eyeball {
        psurf = "EyeBall66.pss" * scale(2.22,0.73,4.68);
        attributes = (eyewhite,navy,eyes);
        centerofmass = (0.00,0.00,2.34);
        mass = 7.85g;
        site base->location = trans(0.00cm,-1.66cm,2.34cm);
        site sight->location = xyz(90.00deg,0.00deg,90.00deg) *
trans(0.00cm,1.01cm,2.34cm);
        site iris->location = trans(0.00cm,-1.66cm,-1.41cm);
        site point->location = trans(0.00cm,-1.66cm,2.45cm);
        site point1->location = xyz(-180.00deg,0.00deg,-180.00deg) *
trans(0.00cm,-1.66cm,3.45cm);
        site LM_pupil_rt->location = trans(0.00cm,0.98cm,2.32cm);
    }
    segment right_foot {
        psurf = "RightFoot66.pss" * scale(19.57,5.03,3.35);
        attribute = shoes;
        centerofmass = (8.61,-1.32,1.68);
        mass = 988.55g;
        site proximal->location = trans(4.35cm,-2.13cm,-1.65cm);
        site distal->location = trans(12.20cm,-2.13cm,5.09cm);
        site normal->location = trans(4.35cm,-2.13cm,5.09cm);
        site new_heel->location = trans(0.01cm,-3.31cm,2.43cm);
        site toes->location = trans(19.33cm,-2.13cm,5.09cm);
        site right->location = trans(18.32cm,4.92cm,3.38cm);
        site left->location = trans(19.28cm,-4.64cm,1.92cm);
        site top->location = trans(1.38cm,4.46cm,-1.01cm);
        site bottom->location = trans(1.35cm,4.45cm,5.09cm);
site heel->location = trans(0.97cm,-2.13cm,5.09cm);
site NS_proximal->location = trans(4.35cm,-2.13cm,-1.65cm);
        site NS_distal->location = trans(4.35cm,-2.13cm,5.00cm);
site LM_pternion->location = trans(0.00cm,-3.32cm,2.43cm);
        site LM_heel_pt_rt_lat->location = trans(0.69cm,-0.08cm,5.10cm);
        site LM_heel_pt_rt_med->location = trans(1.56cm,-5.03cm,5.10cm);
        site LM_fifth_metatarsophalangeal_protrusion->location =
xyz(0.15deg,0.01deg,-0.43deg) * trans(19.10cm,5.03cm,3.60cm);
        site LM_first_metatarsophalangeal_protrusion->location =
trans(19.45cm,-4.75cm,1.97cm);
        site LM_bottom_rfoot->location = trans(4.35cm,-2.13cm,5.09cm);
        site LM_heel->location = trans(0.00cm,-3.33cm,2.43cm);
    }
    segment right_toes {
        psurf = "RightToes66.pss" * scale(7.36,5.03,1.78);
        attribute = shoes;
```

```
        centerofmass = (3.62,0.24,0.27);
        mass = 109.84g;
        site proximal->location = trans(0.22cm,-2.34cm,1.57cm);
        site distal->location = xyz(-44.36deg,90.00deg,-44.36deg) *
trans(1.33cm,-3.41cm,1.49cm);
        site toetip->location = trans(7.39cm,-2.34cm,1.57cm);
        site LM_acropodion->location = trans(7.35cm,0.31cm,-0.19cm);
    }
    segment left_foot {
        psurf = "LeftFoot66.pss" * scale(19.57,5.03,3.35);
        attribute = shoes;
        centerofmass = (8.61,-1.32,1.68);
        mass = 988.55g;
        site proximal->location = trans(4.35cm,2.13cm,-1.65cm);
        site distal->location = trans(12.20cm,2.13cm,5.00cm);
        site normal->location = trans(4.35cm,2.13cm,5.00cm);
        site toes->location = trans(19.33cm,2.13cm,5.00cm);
        site new_heel->location = trans(0.00cm,3.31cm,2.34cm);
        site heel->location = trans(0.97cm,2.13cm,5.00cm);
site NS_proximal->location = trans(4.35cm,2.13cm,-1.65cm);
        site NS_distal->location = trans(4.35cm,2.13cm,5.00cm);
site LM_heel_pt_rt_lat->location = trans(0.69cm,0.08cm,5.10cm);
        site LM_heel_pt_lft_med->location = trans(1.56cm,5.03cm,5.10cm);
        site LM_heel->location = trans(0.00cm,3.33cm,2.43cm);
        site LM_heel_pt_lft_lat->location = trans(0.71cm,0.07cm,5.10cm);
        site LM_pternion->location = trans(0.00cm,3.32cm,2.43cm);
        site LM_fifth_metatarsophalangeal_protrusion->location =
trans(19.10cm,-5.03cm,3.60cm);
        site LM_first_metatarsophalangeal_protrusion->location =
trans(19.45cm,4.75cm,1.97cm);
        site LM_bottom_lfoot->location = trans(4.35cm,2.13cm,5.09cm);
    }
    segment left_toes {
        psurf = "LeftToes66.pss" * scale(7.36,5.03,1.78);
        attribute = shoes;
        centerofmass = (3.62,0.24,0.27);
        mass = 109.84g;
        site proximal->location = trans(0.22cm,2.34cm,1.57cm);
        site distal->location = xyz(-44.36deg,90.00deg,-44.36deg) *
trans(1.33cm,3.41cm,1.49cm);
        site toetip->location = trans(7.39cm,2.34cm,1.57cm);
        site LM_acropodion->location = trans(7.35cm,-0.31cm,-0.19cm);
    }
    segment right_lower_leg {
        psurf = "RtLoLeg66.pss" * scale(6.92,6.01,43.38);
        attribute = pants;
```

```
        centerofmass = (0.32,0.29,19.09);
        mass = 3610.00g;
        site proximal->location = xyz(0.36deg,-0.81deg,0.00deg) *
trans(0.44cm,0.37cm,7.12cm);
        site distal->location = trans(0.00cm,0.00cm,50.51cm);
        site knee->location = trans(6.92cm,0.09cm,7.28cm);
        site back->location = trans(-6.85cm,0.00cm,18.07cm);
        site front->location = trans(5.99cm,0.00cm,16.82cm);
        site right->location = trans(-0.81cm,6.01cm,17.15cm);
        site left->location = trans(-0.86cm,-6.01cm,17.33cm);
        site NS_distal->location = trans(0.00cm,0.00cm,50.51cm);
site NS_proximal->location = xyz(0.36deg,-0.81deg,0.00deg) *
trans(0.44cm,0.37cm,7.12cm);
        site LM_back->location = trans(-6.85cm,0.00cm,18.07cm);
        site LM_front->location = trans(5.99cm,0.00cm,16.82cm);
        site LM_right->location = trans(-0.81cm,6.01cm,17.15cm);
        site LM_left->location = trans(-0.86cm,-6.01cm,17.33cm);
        site LM_calf->location = trans(-0.79cm,6.01cm,17.14cm);
        site LM_midpatella->location = trans(6.55cm,-0.08cm,6.50cm);
        site LM_suprapatella->location = trans(6.74cm,0.13cm,5.28cm);
        site LM_dorsal_juncture->location = trans(0.00cm,0.00cm,0.00cm);
        site LM_lateral_malleolus->location = trans(0.64cm,2.21cm,50.51cm);
        site LM_medial_malleolus->location = trans(0.62cm,-1.92cm,50.51cm);
        site LM_knee_pt_ant->location = trans(6.11cm,0.14cm,4.10cm);
        site LM_ankle_front->location = trans(5.13cm,0.10cm,50.54cm);
        site LM_ankle_back->location = trans(-3.12cm,0.01cm,50.44cm);
        site LM_ankle_left->location = trans(0.62cm,-1.92cm,50.60cm);
        site LM_ankle_right->location = trans(0.64cm,2.21cm,50.55cm);
        site LM_midp_back->location = trans(-5.78cm,0.00cm,8.30cm);
        site LM_midp_left->location = trans(0.97cm,-5.90cm,6.47cm);
        site LM_midp_right->location = trans(1.22cm,5.70cm,6.50cm);
    }
    segment left_lower_leg {
        psurf = "LtLoLeg66.pss" * scale(6.92,6.01,43.38);
        attribute = pants;
        centerofmass = (0.32,0.29,19.09);
        mass = 3610.00g;
        site proximal->location = xyz(-0.36deg,-0.81deg,0.00deg) *
trans(0.44cm,-0.37cm,7.12cm);
        site distal->location = trans(0.00cm,0.00cm,50.51cm);
        site NS_distal->location = trans(0.00cm,0.00cm,50.51cm);
site NS_proximal->location = xyz(-0.36deg,-0.81deg,0.00deg) *
trans(0.44cm,-0.37cm,7.12cm);
        site LM_back->location = trans(-6.85cm,0.00cm,18.07cm);
        site LM_front->location = trans(5.99cm,0.00cm,16.82cm);
        site LM_right->location = trans(-0.81cm,-6.01cm,17.15cm);
```

```
        site LM_left->location = trans(-0.86cm,6.01cm,17.33cm);
        site LM_calf->location = trans(-0.79cm,-6.01cm,17.14cm);
        site LM_midpatella->location = trans(6.55cm,0.08cm,6.50cm);
        site LM_suprapatella->location = trans(6.74cm,-0.13cm,5.28cm);
        site LM_dorsal_juncture->location = trans(0.00cm,0.00cm,0.00cm);
        site LM_lateral_malleolus->location = trans(0.64cm,-2.21cm,50.51cm);
        site LM_medial_malleolus->location = trans(0.62cm,1.92cm,50.51cm);
        site LM_knee_pt_ant->location = trans(6.11cm,-0.14cm,4.10cm);
        site LM_ankle_front->location = trans(5.13cm,-0.10cm,50.54cm);
        site LM_ankle_back->location = trans(-3.12cm,-0.01cm,50.44cm);
        site LM_ankle_left->location = trans(0.62cm,1.92cm,50.60cm);
        site LM_ankle_right->location = trans(0.64cm,-2.21cm,50.55cm);
        site LM_midp_back->location = trans(-5.78cm,0.00cm,8.30cm);
        site LM_midp_left->location = trans(0.97cm,5.90cm,6.47cm);
        site LM_midp_right->location = trans(1.22cm,-5.70cm,6.50cm);
    }
    segment right_upper_leg {
        psurf = "RtUpLeg66.pss" * scale(9.48,8.38,42.64);
        attribute = pants;
        centerofmass = (0.70,0.59,17.48);
        mass = 7850.00g;
        site proximal->location = xyz(0.00deg,3.20deg,0.00deg) *
trans(-1.18cm,1.30cm,2.08cm);
        site distal->location = trans(0.84cm,1.30cm,44.72cm);
        site back->location = trans(-9.47cm,4.36cm,5.96cm);
        site front->location = trans(9.37cm,4.58cm,5.79cm);
        site poplit->location = trans(-5.95cm,0.71cm,44.13cm);
        site right->location = trans(0.80cm,8.34cm,5.95cm);
        site left->location = trans(0.99cm,-8.38cm,5.97cm);
        site poplit2->location = trans(-5.95cm,0.71cm,39.07cm);
        site NS_proximal->location = xyz(0.00deg,3.20deg,0.00deg) *
trans(-1.68cm,1.30cm,7.90cm);
        site NS_distal->location = trans(0.84cm,1.30cm,44.72cm);
        site knee->location = trans(8.74cm,1.23cm,45.35cm);
        site LM_back->location = trans(-9.47cm,4.36cm,5.96cm);
        site LM_front->location = trans(9.37cm,4.58cm,5.79cm);
        site LM_right->location = trans(0.80cm,8.34cm,5.95cm);
        site LM_left->location = trans(0.99cm,-8.38cm,5.97cm);
        site LM_dorsal_juncture->location = trans(-7.38cm,1.93cm,35.91cm);
        site LM_lat_femoral_epicondyle_standing->location =
trans(-1.19cm,5.21cm,42.91cm);
        site LM_popliteal->location = trans(-5.95cm,0.71cm,44.13cm);
        site LM_thigh_back->location = trans(-9.47cm,4.36cm,5.96cm);
        site LM_thigh_front->location = xyz(-0.04deg,0.13deg,0.05deg) *
trans(9.21cm,4.54cm,5.73cm);
        site LM_popliteal_sit->location = trans(-5.95cm,0.71cm,39.07cm);
```

```
    }
    segment left_upper_leg {
        psurf = "LtUpLeg66.pss" * scale(9.48,8.38,42.64);
        attribute = pants;
        centerofmass = (0.70,0.59,17.48);
        mass = 7850.00g;
        site proximal->location = xyz(0.00deg,3.20deg,0.00deg) *
trans(-1.18cm,-1.30cm,2.08cm);
        site distal->location = trans(0.84cm,-1.30cm,44.72cm);
        site NS_proximal->location = xyz(0.00deg,3.20deg,0.00deg) *
trans(-1.68cm,-1.30cm,7.90cm);
site NS_distal->location = trans(0.84cm,-1.30cm,44.72cm);
        site LM_back->location = trans(-9.47cm,-4.36cm,5.96cm);
        site LM_front->location = trans(9.37cm,-4.58cm,5.79cm);
        site LM_right->location = trans(0.80cm,-8.34cm,5.95cm);
        site LM_left->location = trans(0.99cm,8.38cm,5.97cm);
        site LM_dorsal_juncture->location = trans(-7.38cm,-1.93cm,35.91cm);
        site LM_lat_femoral_epicondyle_standing->location =
trans(-1.19cm,5.21cm,42.91cm);
        site LM_popliteal->location = trans(-5.95cm,-0.71cm,44.13cm);
        site LM_thigh_back->location = trans(-9.47cm,-4.36cm,5.96cm);
        site LM_thigh_front->location = xyz(-0.04deg,0.13deg,0.05deg) *
trans(9.21cm,-4.54cm,5.73cm);
        site LM_popliteal_sit->location = trans(-5.95cm,-0.71cm,39.07cm);
    }
    segment upper_torso {
        site proximal->location = trans(0.00cm,0.00cm,0.00cm);
        site distal->location = xyz(0.00deg,16.75deg,0.00deg) *
trans(2.23cm,-0.04cm,-4.16cm);
        site left->location = xyz(180.00deg,0.00deg,0.00deg) *
trans(0.00cm,19.71cm,-3.42cm);
        site right->location = xyz(-180.00deg,0.00deg,0.00deg) *
trans(0.00cm,-19.71cm,-3.42cm);
        site lclav->location = xyz(-90.00deg,0.00deg,-5.00deg) *
trans(2.50cm,7.57cm,-5.39cm);
        site rclav->location = xyz(90.00deg,0.00deg,5.00deg) *
trans(2.50cm,-7.57cm,-5.39cm);
    }
    segment bottom_head {
        psurf = "Head66.pss" * scale(10.55,8.79,13.10);
        attributes = (flesh,upper_lip,hair,lower_lip);
        centerofmass = (-0.18,0.09,10.87);
        mass = 6200.00g;
        site right_eyeball->location = xyz(90.00deg,0.00deg,-90.00deg) *
trans(4.86cm,-2.86cm,11.17cm);
        site left_eyeball->location = xyz(90.00deg,0.00deg,-90.00deg) *
```

```
trans(4.86cm,2.86cm,11.17cm);
        site proximal->location = trans(2.09cm,0.00cm,10.04cm);
        site sight->location = xyz(44.96deg,-90.00deg,-44.96deg) *
trans(9.87cm,0.00cm,11.17cm);
        site top->location = trans(2.09cm,0.00cm,23.16cm);
        site eye_level->location = trans(-4.48cm,-10.41cm,11.17cm);
        site eye_lvl_out->location = trans(-20.02cm,0.00cm,11.17cm);
        site top_out->location = trans(-20.02cm,0.00cm,23.16cm);
        site bottom->location = trans(-4.48cm,-10.34cm,0.78cm);
        site back->location = trans(-10.55cm,0.01cm,14.55cm);
        site front->location = trans(8.70cm,0.06cm,13.84cm);
        site right->location = trans(1.70cm,-7.25cm,9.80cm);
        site left->location = trans(1.70cm,7.25cm,9.80cm);
        site NS_distal->location = trans(2.09cm,0.00cm,23.16cm);
        site NS_proximal->location = trans(2.09cm,0.00cm,10.04cm);
        site top_side->location = trans(-5.00cm,-10.41cm,23.16cm);
        site LM_right->location = trans(1.70cm,-7.25cm,9.80cm);
        site LM_left->location = trans(1.70cm,7.25cm,9.80cm);
        site LM_back->location = trans(-10.55cm,0.01cm,14.55cm);
        site LM_chin->location = trans(7.76cm,0.07cm,1.11cm);
        site LM_glabella->location = trans(8.71cm,0.06cm,13.83cm);
        site LM_frontotemporale_r->location = trans(5.83cm,-6.03cm,13.07cm);
        site LM_frontotemporale_l->location = trans(5.83cm,6.18cm,13.07cm);
        site LM_ectocanthus->location = trans(6.58cm,-4.67cm,11.17cm);
        site LM_alare_r->location = trans(8.38cm,-1.69cm,7.17cm);
        site LM_alare_l->location = trans(8.38cm,1.69cm,7.17cm);
        site LM_cheilion_r->location = trans(7.15cm,-3.24cm,3.84cm);
        site LM_cheilion_l->location = trans(7.15cm,3.38cm,4.26cm);
        site LM_crinion->location = trans(7.72cm,0.02cm,16.91cm);
        site LM_ear_top->location = trans(1.72cm,-8.21cm,12.47cm);
        site LM_ear_pt->location = trans(-0.16cm,-8.73cm,10.89cm);
        site LM_ear_bottom->location = trans(1.19cm,-7.36cm,7.15cm);
        site LM_infraorbitale_r->location = trans(5.79cm,-2.90cm,8.95cm);
        site LM_infraorbitale_l->location = trans(5.79cm,2.90cm,8.95cm);
        site LM_menton->location = trans(6.19cm,0.12cm,0.00cm);
        site LM_pronasale->location = trans(10.40cm,0.06cm,7.72cm);
        site LM_promenton->location = trans(7.77cm,0.07cm,1.11cm);
        site LM_otobasion_sup->location = trans(2.72cm,-6.93cm,11.41cm);
        site LM_sellion->location = trans(8.30cm,0.06cm,10.90cm);
        site LM_stomion->location = trans(8.34cm,0.06cm,3.91cm);
        site LM_head_top->location = trans(0.74cm,0.09cm,23.04cm);
        site LM_top_of_head->location = trans(-4.51cm,0.12cm,23.13cm);
        site LM_tragion_rt->location = trans(2.06cm,-7.45cm,10.06cm);
        site LM_tragion_lft->location = trans(2.06cm,7.45cm,10.06cm);
        site LM_ectoorbitale_rt->location = trans(4.93cm,-6.35cm,12.01cm);
        site LM_ectoorbitale_lft->location = trans(4.93cm,6.35cm,12.01cm);
```

```
        site LM_subnasale->location = trans(8.56cm,0.05cm,6.17cm);
        site LM_gonion_rt->location = trans(0.50cm,-5.79cm,2.78cm);
        site LM_gonion_lft->location = trans(0.50cm,5.79cm,2.78cm);
        site LM_zygion_rt->location = trans(5.39cm,-5.90cm,9.97cm);
        site LM_zygion_lft->location = trans(5.39cm,5.90cm,9.97cm);
        site LM_zygofrontale_rt->location = trans(6.29cm,-5.13cm,13.73cm);
        site LM_zygofrontale_lft->location = trans(6.29cm,5.13cm,13.73cm);
        site LM_TOP_HEAD->location = trans(2.09cm,0.00cm,23.16cm);
        site LM_TOP_HEAD_SIDE->location = trans(-5.00cm,-10.41cm,23.16cm);
        site LM_TOP_HEAD_BACK->location = trans(-20.02cm,0.00cm,23.16cm);
    }
    segment neck {
        psurf = "Neck66.pss" * scale(7.22,7.22,10.58);
        attribute = flesh;
        centerofmass = (0.29,0.03,8.18);
        mass = 156.91g;
        site proximal->location = xyz(0.00deg,8.03deg,0.00deg) *
trans(-1.19cm,-0.04cm,8.43cm);
        site distal->location = trans(4.53cm,-0.33cm,19.01cm);
        site front->location = xyz(0.00deg,-18.64deg,0.00deg) *
trans(6.12cm,0.00cm,7.20cm);
        site back->location = trans(-6.52cm,-0.04cm,9.00cm);
        site base_rt->location = trans(-1.16cm,-7.22cm,7.46cm);
        site proximal0->location = trans(-1.19cm,-0.04cm,8.43cm);
        site distal0->location = trans(-1.19cm,-0.04cm,19.01cm);
        site LM_front->location = trans(6.12cm,0.00cm,7.20cm);
        site LM_back->location = trans(-6.66cm,-0.04cm,7.19cm);
        site LM_cervicale->location = trans(-5.92cm,0.06cm,11.73cm);
        site LM_infrathyroid->location = trans(6.08cm,0.00cm,7.33cm);
        site LM_submandibular->location = trans(4.33cm,-0.26cm,9.16cm);
        site LM_neck_rt_lat->location = trans(5.15cm,-5.00cm,4.55cm);
        site LM_neck_lft_lat->location = trans(5.15cm,5.00cm,4.55cm);
        site LM_trapezius_pt_rt->location = trans(-2.01cm,-5.32cm,9.38cm);
        site LM_trapezius_pt_lft->location = trans(-2.01cm,5.32cm,9.38cm);
        site LM_cervicale0->location = xyz(-0.51deg,-21.26deg,-0.10deg) *
trans(-6.87cm,-0.01cm,3.96cm);
    }
    segment right_clavicle {
        psurf = "RtClavicle66.pss" * scale(5.30,10.60,22.50);
        attribute = shirt;
        centerofmass = (-0.57,-1.93,11.84);
        mass = 2350.00g;
        site proximal->location = trans(0.00cm,5.02cm,7.00cm);
        site lateral->location = xyz(90.00deg,0.00deg,0.00deg) *
trans(0.00cm,2.99cm,19.11cm);
        site shoulder_back->location = trans(-11.42cm,3.65cm,20.57cm);
```

```
        site top->location = trans(-1.02cm,4.86cm,16.72cm);
        site LM_clavicle_pt_r->location = trans(2.97cm,6.93cm,18.81cm);
        site LM_midshoulder->location = trans(-1.02cm,9.17cm,11.11cm);
        site LM_clavicle_pt_rt->location = trans(-1.01cm,7.98cm,16.72cm);
        site LM_post_scye_rt->location = trans(-2.89cm,-6.61cm,10.84cm);
        site LM_midscye_rt->location = xyz(-180.00deg,0.00deg,-0.02deg) *
trans(-5.33cm,0.05cm,16.45cm);
    }
    segment left_clavicle {
        psurf = "LtClavicle66.pss" * scale(5.30,10.60,22.50);
        attribute = shirt;
        centerofmass = (-0.57,-1.93,11.84);
        mass = 2350.00g;
        site proximal->location = trans(0.00cm,-5.02cm,7.00cm);
        site lateral->location = xyz(-90.00deg,0.00deg,0.00deg) *
trans(0.00cm,-2.99cm,19.11cm);
        site LM_clavicle_pt_l->location = trans(2.99cm,-6.93cm,18.78cm);
        site LM_clavicle_pt_lft->location = trans(-1.04cm,-7.98cm,16.72cm);
        site LM_post_scye_lft->location = trans(-2.89cm,6.61cm,10.84cm);
        site LM_midscye_lft->location = xyz(-180.00deg,0.00deg,-0.02deg) *
trans(-5.33cm,0.05cm,16.45cm);
    }
    segment right_upper_arm {
        psurf = "RightUpArm66.pss";
        attribute = shirt;
        centerofmass = (0.00,0.00,0.48);
        mass = 2200.00g;
        site proximal->location = trans(0.15cm,0.09cm,0.11cm);
site distal->location = trans(0.00cm,0.09cm,1.00cm);
        site shoulder_level->location = trans(0.10cm,-1.45cm,0.00cm);
        site shlder_lvl_out->location = xyz(180.00deg,-5.82deg,0.02deg) *
trans(-3.14cm,-2.89cm,0.00cm);
        site deltoid->location = trans(-0.07cm,0.82cm,0.09cm);
        site right->location = trans(-0.01cm,1.00cm,0.59cm);
        site left->location = trans(-0.01cm,-1.00cm,0.59cm);
        site front->location = trans(0.14cm,0.02cm,0.03cm);
site back->location = trans(-0.98cm,0.57cm,0.55cm);
        site NS_proximal->location = trans(-0.10cm,0.10cm,0.00cm);
        site NS_distal->location = trans(0.00cm,0.09cm,1.00cm);
        site LM_shoulder_level->location = trans(0.10cm,-1.45cm,0.00cm);
        site LM_shlder_lvl_out->location = xyz(180.00deg,-5.82deg,0.02deg) *
trans(-3.14cm,-2.89cm,0.00cm);
        site LM_right->location = trans(-0.01cm,1.00cm,0.59cm);
        site LM_left->location = trans(-0.01cm,-1.00cm,0.59cm);
        site LM_front->location = trans(0.95cm,0.00cm,0.55cm);
        site LM_back->location = trans(-0.98cm,0.57cm,0.55cm);
```

```
        site LM_deltoid->location = trans(-0.07cm,0.82cm,0.09cm);
site LM_acromion_r->location = trans(-0.10cm,0.10cm,0.00cm);
        site LM_biceps_pt->location = trans(1.20cm,0.57cm,0.63cm);
        site LM_radiale->location = trans(-0.04cm,0.85cm,1.00cm);
        site LM_ant_scye_upper_arm->location =
xyz(174.23deg,87.22deg,95.70deg) * trans(0.80cm,-0.38cm,0.33cm);
        site LM_deltoid_pt_rt->location = trans(-0.04cm,0.84cm,0.25cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(5.34,5.34,34.08),
(6.14,6.86,34.08),"z");
    }
    segment left_upper_arm {
        psurf = "LeftUpArm66.pss";
        attribute = shirt;
        centerofmass = (0.00,0.00,0.48);
        mass = 2200.00g;
site proximal->location = trans(0.15cm,-0.09cm,0.11cm);
site distal->location = trans(0.00cm,-0.09cm,1.00cm);
site NS_proximal->location = trans(-0.10cm,-0.10cm,0.00cm);
site NS_distal->location = trans(0.00cm,-0.09cm,1.00cm);
        site deltoid->location = trans(-0.07cm,-0.81cm,0.09cm);
        site LM_deltoid->location = trans(-0.07cm,-0.81cm,0.09cm);
site LM_acromion_l->location = trans(-0.10cm,-0.10cm,0.00cm);
        site LM_deltoid_pt_lft->location = trans(-0.04cm,-0.84cm,0.25cm);
        site LM_shoulder_level->location = trans(0.10cm,1.45cm,0.00cm);
        site LM_shlder_lvl_out->location = xyz(180.00deg,-5.82deg,0.02deg) *
trans(-3.14cm,2.89cm,0.00cm);
        site LM_right->location = trans(-0.01cm,-1.00cm,0.59cm);
        site LM_left->location = trans(-0.01cm,1.00cm,0.59cm);
        site LM_front->location = trans(0.14cm,-0.02cm,0.03cm);
        site LM_back->location = trans(-0.98cm,-0.57cm,0.55cm);
        site LM_biceps_pt->location = trans(1.20cm,-0.57cm,0.63cm);
        site LM_radiale->location = trans(-0.04cm,-0.85cm,1.00cm);
        site LM_ant_scye_upper_arm->location =
xyz(174.23deg,87.22deg,95.70deg) * trans(0.80cm,0.38cm,0.33cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(5.34,5.34,34.08),
(6.14,6.86,34.08),"z");
    }
    segment right_lower_arm {
        psurf = "RightLoArm66.pss";
        attribute = flesh;
        centerofmass = (0.03,0.02,0.41);
        mass = 1260.00g;
        site proximal->location = trans(0.22cm,0.10cm,0.13cm);
        site distal->location = trans(0.22cm,0.10cm,1.13cm);
        site elbow->location = trans(-0.86cm,0.39cm,0.06cm);
        site right->location = trans(-0.04cm,0.97cm,0.26cm);
```

```
        site left->location = trans(0.00cm,-0.93cm,0.26cm);
        site front->location = trans(0.98cm,0.53cm,0.26cm);
site back->location = trans(-1.00cm,0.49cm,0.26cm);
site NS_proximal->location = trans(0.22cm,0.10cm,0.13cm);
site NS_distal->location = trans(0.22cm,0.10cm,1.13cm);
        site LM_right->location = trans(-0.04cm,0.97cm,0.26cm);
        site LM_left->location = trans(0.00cm,-0.93cm,0.26cm);
        site LM_front->location = trans(0.98cm,0.53cm,0.26cm);
        site LM_back->location = trans(-1.00cm,0.49cm,0.26cm);
        site LM_elbow->location = trans(-0.86cm,0.39cm,0.06cm);
site LM_elbow_crease->location = xyz(-0.29deg,6.13deg,-0.03deg) *
trans(0.98cm,0.00cm,0.29cm);
        site LM_stylion->location = trans(0.47cm,-0.16cm,1.15cm);
        site LM_olecranon_ext->location = trans(-0.93cm,0.10cm,0.13cm);
        site LM_olecranon_flex->location = trans(-0.86cm,0.12cm,0.06cm);
        site LM_styl_front->location = trans(0.47cm,0.10cm,1.15cm);
        site LM_styl_back->location = trans(-0.27cm,0.10cm,1.15cm);
        site LM_styl_left->location = trans(0.08cm,-0.42cm,1.15cm);
        site LM_styl_right->location = trans(0.08cm,0.61cm,1.15cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(4.09,4.57,26.95),
(6.6,3.57,26.95),"z");
    }
    segment left_lower_arm {
        psurf = "LeftLoArm66.pss";
        attribute = flesh;
        centerofmass = (0.03,0.02,0.41);
        mass = 1260.00g;
        site proximal->location = trans(0.22cm,-0.10cm,0.13cm);
        site distal->location = trans(0.22cm,-0.10cm,1.13cm);
        site NS_proximal->location = trans(0.22cm,-0.10cm,0.13cm);
site NS_distal->location = trans(0.22cm,-0.10cm,1.13cm);
        site LM_left->location = trans(0.00cm,0.93cm,0.26cm);
        site LM_front->location = trans(0.98cm,-0.53cm,0.26cm);
        site LM_back->location = trans(-1.00cm,-0.49cm,0.26cm);
        site LM_elbow->location = trans(-0.86cm,-0.39cm,0.06cm);
        site LM_elbow_crease->location = xyz(-0.29deg,6.13deg,-0.03deg) *
trans(0.98cm,0.00cm,0.29cm);
        site LM_stylion->location = trans(0.47cm,0.16cm,1.15cm);
        site LM_olecranon_ext->location = trans(-0.93cm,-0.10cm,0.13cm);
        site LM_olecranon_flex->location = trans(-0.86cm,-0.12cm,0.06cm);
        site LM_styl_front->location = trans(0.47cm,-0.10cm,1.15cm);
        site LM_styl_back->location = trans(-0.27cm,-0.10cm,1.15cm);
        site LM_styl_left->location = trans(0.08cm,0.42cm,1.15cm);
        site LM_styl_right->location = trans(0.08cm,-0.61cm,1.15cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(4.09,4.57,26.95),
(6.6,3.57,26.95),"z");
```

```
    }
    segment t1 {
        psurf = "Spine_T166.pss" * scale(13.93,2.01,8.41);
        attribute = shirt;
        centerofmass = (0.00,1.09,1.76);
        mass = 1440.00g;
        site cy2->location = xyz(-117.16deg,0.00deg,-90.00deg) *
trans(0.00cm,2.11cm,-5.29cm);
        site proximal->location = trans(0.00cm,0.00cm,-5.29cm);
        site distal->location = xyz(-109.50deg,0.00deg,-90.00deg) *
trans(0.00cm,2.01cm,-5.29cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-5.29cm);
site NS_distal->location = xyz(-109.50deg,0.00deg,-90.00deg) *
trans(0.00cm,2.01cm,-5.29cm);
site LM_neck_ant->location = trans(0.04cm,1.93cm,4.24cm);
        site LM_suprasternale->location = trans(0.00cm,2.01cm,3.75cm);
    }
    segment t2 {
        psurf = "Spine_T266.pss" * scale(16.17,2.13,10.12);
        attribute = shirt;
        centerofmass = (-0.01,1.28,-1.57);
        mass = 1440.00g;
        site front->location = trans(0.00cm,0.00cm,10.14cm);
        site back->location = trans(0.00cm,0.02cm,-10.11cm);
        site proximal->location = trans(0.00cm,0.00cm,-7.32cm);
        site distal->location = xyz(-1.53deg,0.00deg,0.00deg) *
trans(0.00cm,2.13cm,-7.32cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-7.32cm);
        site NS_distal->location = xyz(-1.53deg,0.00deg,0.00deg) *
trans(0.00cm,2.13cm,-7.32cm);
    }
    segment t3 {
        psurf = "Spine_T366.pss" * scale(16.74,1.98,10.80);
        attribute = shirt;
        centerofmass = (-0.02,1.24,-0.85);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-8.09cm);
        site distal->location = xyz(1.58deg,0.00deg,0.00deg) *
trans(0.00cm,1.98cm,-8.09cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-8.09cm);
site NS_distal->location = xyz(1.58deg,0.00deg,0.00deg) *
trans(0.00cm,1.98cm,-8.09cm);
    }
    segment t4 {
        psurf = "Spine_T466.pss" * scale(17.06,2.83,10.47);
        attribute = shirt;
```

```
        centerofmass = (0.02,2.63,-0.41);
        mass = 1440.00g;
        site interscye_left->location = trans(17.06cm,0.00cm,-0.20cm);
        site interscye_right->location = trans(-17.06cm,0.00cm,-0.19cm);
site front->location = trans(0.00cm,-1.68cm,10.56cm);
        site back->location = trans(0.00cm,0.01cm,-10.46cm);
        site proximal->location = trans(0.00cm,0.00cm,-7.97cm);
        site distal->location = xyz(3.64deg,0.00deg,0.00deg) *
trans(0.00cm,2.83cm,-7.97cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-7.97cm);
site NS_distal->location = xyz(3.64deg,0.00deg,0.00deg) *
trans(0.00cm,2.83cm,-7.97cm);
site LM_back_pt_rt->location = trans(-11.75cm,1.42cm,-5.41cm);
site LM_back_pt_lft->location = trans(11.75cm,1.42cm,-5.41cm);
site LM_interscye_left->location = trans(17.06cm,0.00cm,-0.20cm);
        site LM_interscye_right->location = trans(-17.06cm,0.00cm,-0.19cm);

    }
    segment t5 {
        psurf = "Spine_T566.pss" * scale(16.54,2.29,11.31);
        attribute = shirt;
        centerofmass = (-0.06,1.52,-0.56);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-8.71cm);
        site distal->location = xyz(3.95deg,0.00deg,0.00deg) *
trans(0.00cm,2.29cm,-8.71cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-8.71cm);
        site NS_distal->location = xyz(3.95deg,0.00deg,0.00deg) *
trans(0.00cm,2.29cm,-8.71cm);
    }
    segment t6 {
        psurf = "Spine_T666.pss" * scale(16.54,2.42,11.39);
        attribute = shirt;
        centerofmass = (0.00,1.65,-0.15);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-8.81cm);
        site distal->location = xyz(4.78deg,0.00deg,0.00deg) *
trans(0.00cm,2.42cm,-8.81cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-8.81cm);
        site NS_distal->location = xyz(4.78deg,0.00deg,0.00deg) *
trans(0.00cm,2.42cm,-8.81cm);
    }
    segment t7 {
        psurf = "Spine_T766.pss" * scale(16.54,2.42,11.59);
        attribute = shirt;
        centerofmass = (-0.02,1.73,0.00);
```

```
        mass = 1440.00g;
        site front->location = trans(0.00cm,1.34cm,12.95cm);
        site back->location = trans(0.00cm,1.34cm,-11.59cm);
        site right->location = trans(-16.54cm,2.56cm,0.48cm);
        site left->location = trans(16.54cm,2.56cm,0.48cm);
        site proximal->location = trans(0.00cm,0.13cm,-8.87cm);
        site distal->location = xyz(5.26deg,0.00deg,0.00deg) *
trans(0.00cm,2.56cm,-8.87cm);
site NS_proximal->location = trans(0.00cm,0.13cm,-8.87cm);
site NS_distal->location = xyz(5.26deg,0.00deg,0.00deg) *
trans(0.00cm,2.56cm,-8.87cm);
site LM_chest_right->location = trans(-16.55cm,2.56cm,0.49cm);
        site LM_chest_left->location = trans(16.54cm,2.56cm,0.48cm);
        site LM_chest_back->location = trans(0.00cm,1.34cm,-11.59cm);
        site LM_chest_front->location = trans(0.00cm,1.34cm,11.39cm);
    }
    segment t8 {
        psurf = "Spine_T866.pss" * scale(16.54,2.42,11.59);
        attribute = shirt;
        centerofmass = (-0.04,1.81,-0.08);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.22cm,-8.89cm);
        site distal->location = xyz(4.73deg,0.00deg,0.00deg) *
trans(0.00cm,2.66cm,-8.89cm);
site NS_proximal->location = trans(0.00cm,0.22cm,-8.89cm);
        site NS_distal->location = xyz(4.73deg,0.00deg,0.00deg) *
trans(0.00cm,2.66cm,-8.89cm);
site LM_midspine->location = trans(0.00cm,1.44cm,-11.58cm);
        site LM_midspine0->location = trans(0.00cm,0.01cm,-11.56cm);
        site LM_ant_scye_torso->location = trans(-16.54cm,2.64cm,0.31cm);
    }
    segment t9 {
        psurf = "Spine_T966.pss" * scale(16.54,2.56,11.65);
        attribute = shirt;
        centerofmass = (0.11,1.87,-0.19);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-8.95cm);
        site distal->location = xyz(4.58deg,0.00deg,0.00deg) *
trans(0.00cm,2.56cm,-8.95cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-8.95cm);
site NS_distal->location = xyz(4.58deg,0.00deg,0.00deg) *
trans(0.00cm,2.56cm,-8.95cm);
    }
    segment t10 {
        psurf = "Spine_T1066.pss" * scale(16.54,2.78,11.56);
        attribute = shirt;
```

```
        centerofmass = (0.02,1.92,-0.16);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-8.82cm);
        site distal->location = xyz(1.58deg,0.00deg,0.00deg) *
trans(0.00cm,2.78cm,-8.82cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-8.82cm);
        site NS_distal->location = xyz(1.58deg,0.00deg,0.00deg) *
trans(0.00cm,2.78cm,-8.82cm);
    }
    segment t11 {
        psurf = "Spine_T1166.pss" * scale(16.11,1.80,11.30);
        attribute = shirt;
        centerofmass = (0.00,1.35,-0.01);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.17cm,-8.45cm);
        site distal->location = trans(0.00cm,1.97cm,-8.45cm);
site NS_proximal->location = trans(0.00cm,0.17cm,-8.45cm);
site NS_distal->location = trans(0.00cm,1.97cm,-8.45cm);
    }
    segment t12 {
        psurf = "Spine_T1266.pss" * scale(16.11,3.90,11.00);
        attribute = shirt;
        centerofmass = (0.01,2.86,-0.08);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.23cm,-8.06cm);
        site distal->location = xyz(-3.07deg,0.00deg,0.00deg) *
trans(0.00cm,4.14cm,-8.06cm);
site NS_proximal->location = trans(0.00cm,0.23cm,-8.06cm);
        site NS_distal->location = xyz(-3.07deg,0.00deg,0.00deg) *
trans(0.00cm,4.14cm,-8.06cm);
    }
    segment l1 {
        psurf = "Spine_L166.pss" * scale(16.11,3.66,10.59);
        attribute = shirt;
        centerofmass = (0.00,2.49,0.00);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site distal->location = trans(0.00cm,3.66cm,-7.55cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site NS_distal->location = trans(0.00cm,3.66cm,-7.55cm);
    }
    segment l2 {
        psurf = "Spine_L266.pss" * scale(15.82,3.66,10.59);
        attribute = shirt;
        centerofmass = (0.00,2.49,0.00);
        mass = 1440.00g;
```

```
        site proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site distal->location = trans(0.00cm,3.66cm,-7.55cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site NS_distal->location = trans(0.00cm,3.66cm,-7.55cm);
    }
    segment 13 {
        psurf = "Spine_L366.pss" * scale(15.47,2.70,10.59);
        attribute = shirt;
        centerofmass = (0.00,1.83,0.00);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site distal->location = trans(0.00cm,2.70cm,-7.55cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site NS_distal->location = trans(0.00cm,2.70cm,-7.55cm);
site LM_tenth_rib->location = trans(-9.54cm,0.16cm,8.12cm);
    }
    segment 14 {
        psurf = "Spine_L466.pss" * scale(15.47,3.51,10.59);
        attribute = shirt;
        centerofmass = (0.00,2.38,0.00);
        mass = 1440.00g;
        site proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site distal->location = trans(0.00cm,3.51cm,-7.55cm);
site NS_proximal->location = trans(0.00cm,0.00cm,-7.55cm);
        site NS_distal->location = trans(0.00cm,3.51cm,-7.55cm);
site LM_waist_rt->location = trans(-15.47cm,2.09cm,0.00cm);
        site LM_waist_lft->location = trans(15.47cm,2.09cm,0.00cm);
        site LM_waist_ant_navel->location = trans(0.00cm,0.09cm,10.59cm);
        site LM_waist_post->location = trans(-0.06cm,2.09cm,-9.57cm);
    }
    segment 15 {
        psurf = "Spine_L566.pss" * scale(15.47,3.57,10.94);
        attribute = shirt;
        centerofmass = (0.05,2.45,-0.50);
        mass = 1440.00g;
        site right->location = trans(-15.03cm,3.74cm,-0.84cm);
        site left->location = trans(15.07cm,3.75cm,-0.83cm);
        site front->location = trans(-7.52cm,3.74cm,8.89cm);
        site back->location = trans(-7.50cm,3.74cm,-10.59cm);
        site proximal->location = xyz(-90.00deg,0.00deg,-90.00deg) *
trans(0.01cm,0.17cm,-7.78cm);
        site distal->location = trans(0.01cm,3.74cm,-7.77cm);
site NS_proximal->location = xyz(-90.00deg,0.00deg,-90.00deg) *
trans(0.01cm,0.17cm,-7.78cm);
site NS_distal->location = trans(0.01cm,3.74cm,-7.77cm);
site LM_waist_lft->location = trans(15.25cm,3.74cm,-0.37cm);
```

```
        site LM_waist_ant->location = trans(0.01cm,3.74cm,10.02cm);
        site LM_waist_rt->location = trans(-15.24cm,3.74cm,-0.37cm);
        site LM_iliocristale->location = trans(-13.81cm,0.40cm,3.47cm);
    }
    segment lower_torso {
        psurf = "LowerTorso66.pss" * scale(11.70,17.06,14.53);
        attribute = pants;
        centerofmass = (5.85,8.53,13.07);
        mass = 9890.00g;
        site proximal->location = trans(2.58cm,0.00cm,3.01cm);
        site distal->location = trans(-7.28cm,0.00cm,25.34cm);
        site floor->location = xyz(42.08deg,90.00deg,42.08deg) *
trans(2.58cm,0.00cm,-106.20cm);
        site ldistal->location = xyz(-90.00deg,0.00deg,0.00deg) *
trans(2.58cm,0.00cm,3.01cm);
        site rdistal->location = xyz(90.00deg,0.00deg,0.00deg) *
trans(2.58cm,0.00cm,3.01cm);
        site seat->location = trans(2.58cm,0.00cm,-9.81cm);
        site rhip_lateral->location = xyz(-180.00deg,0.00deg,0.00deg) *
trans(0.00cm,-9.56cm,10.81cm);
        site lhip_lateral->location = xyz(-180.00deg,0.00deg,0.00deg) *
trans(0.00cm,9.56cm,10.81cm);
        site center_of_mass->location = trans(1.68cm,1.14cm,25.29cm);
        site left_hiphandle->location = xyz(108.21deg,-20.58deg,-102.00deg) *
trans(0.69cm,17.75cm,20.77cm);
        site right_hiphandle->location = xyz(-108.21deg,-20.58deg,-67.00deg) *
 trans(-0.37cm,-18.76cm,22.01cm);
        site butt->location = trans(-11.94cm,-12.56cm,8.67cm);
        site crotch_level->location = trans(2.57cm,-10.15cm,3.01cm);
        site left_side->location = trans(2.00cm,17.06cm,12.59cm);
        site right_side->location = trans(1.99cm,-17.06cm,12.59cm);
        site sit_ext->location = trans(-21.12cm,0.00cm,4.98cm);
        site sit_level->location = trans(-21.12cm,0.00cm,4.98cm);
        site front->location = trans(10.52cm,-12.55cm,8.68cm);
        site NS_proximal->location = trans(2.57cm,0.00cm,10.81cm);
site NS_distal->location = xyz(0.00deg,90.00deg,0.00deg) *
trans(2.59cm,0.00cm,25.34cm);
        site LM_hip_joint_rt->location = trans(0.00cm,-9.56cm,10.81cm);
        site LM_hip_joint_lft->location = trans(0.00cm,9.56cm,10.81cm);
        site LM_buttock_pt_post->location = trans(-11.59cm,0.00cm,13.68cm);
        site LM_buttock_pt_rt_lat->location = trans(-2.94cm,-15.41cm,11.40cm);
        site LM_buttock_pt_lft_lat->location = trans(-2.94cm,15.41cm,11.40cm);
        site LM_gluteal_furrow_pt2_rt->location =
trans(-8.15cm,-8.97cm,3.08cm);
        site LM_gluteal_furrow_pt2_lft->location =
trans(-8.15cm,8.97cm,3.08cm);
```

```
        site LM_ant_sup_iliac_spine_rt->location =
trans(10.99cm,-11.70cm,16.19cm);
        site LM_ant_sup_iliac_spine_lft->location =
trans(10.99cm,11.70cm,16.19cm);
        site LM_post_iliac_spine_rt->location =
trans(-11.08cm,-3.44cm,21.96cm);
        site LM_post_iliac_spine_lft->location =
trans(-11.08cm,3.44cm,21.96cm);
        site LM_trochanter_rt->location = trans(0.07cm,-16.31cm,10.01cm);
        site LM_trochanterion_rt->location = trans(2.46cm,-16.94cm,7.43cm);
        site LM_trochanter_lft->location = trans(0.07cm,16.31cm,10.01cm);
        site LM_trochanterion_lft->location = trans(2.46cm,16.94cm,7.43cm);
        site LM_front_pelvis->location = trans(11.03cm,0.01cm,13.68cm);
        site LM_crotch_level_rt->location = trans(2.57cm,-10.15cm,3.01cm);
        site LM_crotch_level_lft->location = trans(2.57cm,10.15cm,3.01cm);
        site LM_crotch->location = trans(2.57cm,0.22cm,3.01cm);
        site LM_sit_level->location = trans(-21.12cm,0.00cm,4.98cm);
    }
    segment left_palm {
        psurf = "lpalm.pss" * scale(4.76,8.45,1.65);
        attribute = flesh;
        centerofmass = (0.55,3.72,0.07);
        mass = 313.82g;
        site NS_proximal->location = xyz(-90.00deg,6.56deg,-180.00deg) *
trans(-1.62cm,0.00cm,0.02cm);
site NS_distal->location = xyz(177.02deg,-0.14deg,-175.13deg) *
trans(-1.04cm,8.45cm,-0.06cm);
site base->location = xyz(-90.00deg,6.56deg,-180.00deg) *
trans(-1.62cm,0.00cm,0.02cm);
site f11->location = xyz(177.02deg,-0.14deg,-175.13deg) *
trans(-3.52cm,8.80cm,0.01cm);
        site f22->location = xyz(177.02deg,-0.14deg,-175.13deg) *
trans(-1.04cm,8.45cm,-0.06cm);
site f33->location = xyz(177.02deg,-0.14deg,-175.13deg) *
trans(1.58cm,8.14cm,0.20cm);
        site f44->location = xyz(177.02deg,-0.14deg,-175.13deg) *
trans(3.92cm,7.32cm,0.48cm);
        site left_bird->location = xyz(-94.00deg,6.39deg,-179.88deg) *
trans(0.01cm,7.35cm,-0.69cm);
        site palmcenter->location = trans(0.19cm,5.77cm,-0.08cm);
        site thumb0->location = xyz(-2.98deg,0.14deg,15.13deg) *
trans(-1.85cm,1.96cm,0.62cm);
        site LM_hand_closed_center->location = trans(-0.19cm,7.77cm,-0.08cm);
    }
    segment left_finger32 {
        psurf = "lpinfinger02.pss" * scale(0.87,2.72,0.78);
```

```
        attribute = flesh;
        centerofmass = (0.02,1.91,-0.04);
        mass = 2.54g;
        site base0->location = trans(0.00cm,0.68cm,-0.08cm);
        site tip->location = trans(0.07cm,3.40cm,0.44cm);
site NS_proximal->location = trans(0.00cm,0.68cm,-0.08cm);
        site NS_distal->location = trans(0.07cm,3.40cm,0.44cm);
    }
    segment left_finger31 {
        psurf = "lpinfinger01.pss";
        attribute = flesh;
        centerofmass = (-0.04,0.65,-0.01);
        mass = 2.17g;
        site NS_proximal->location = trans(-0.06cm,0.33cm,-0.03cm);
site NS_distal->location = trans(-0.06cm,1.33cm,-0.01cm);
site base0->location = trans(-0.06cm,0.33cm,-0.03cm);
        site tip->location = trans(-0.06cm,1.33cm,-0.01cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(0.95,1.75,0.92),
(1.04,1.75,0.94),"y");
    }
    segment left_finger30 {
        psurf = "lpinfinger00.pss";
        attribute = flesh;
        centerofmass = (-0.09,0.91,0.10);
        mass = 9.94g;
        site base0->location = trans(-0.23cm,0.28cm,-0.09cm);
site tip->location = trans(-0.23cm,1.28cm,-0.09cm);
site NS_proximal->location = trans(-0.23cm,0.28cm,-0.09cm);
site NS_distal->location = trans(-0.23cm,1.28cm,-0.09cm);
site LM_metacarpale_V->location = trans(-0.76cm,0.24cm,-0.12cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(1.24,4.14,0.9),
(1.53,4.14,1.47),"y");
    }
    segment left_finger22 {
        psurf = "lringfinger02.pss" * scale(0.92,2.97,0.86);
        attribute = flesh;
        centerofmass = (0.02,2.06,-0.11);
        mass = 3.14g;
        site base0->location = trans(-0.01cm,0.60cm,0.09cm);
        site tip->location = trans(0.04cm,3.57cm,0.47cm);
site NS_proximal->location = trans(-0.01cm,0.60cm,0.09cm);
        site NS_distal->location = trans(0.04cm,3.57cm,0.47cm);
    }
    segment left_finger21 {
        psurf = "lringfinger01.pss";
        attribute = flesh;
```

```
        centerofmass = (0.03,0.59,0.10);
        mass = 3.55g;
        site base0->location = trans(0.02cm,0.25cm,-0.06cm);
        site tip->location = trans(0.01cm,1.25cm,-0.07cm);
site NS_proximal->location = trans(0.02cm,0.25cm,-0.06cm);
        site NS_distal->location = trans(0.01cm,1.25cm,-0.07cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.07,2.42,1.03),
(1.1,2.42,1.03),"y");
    }
    segment left_finger20 {
        psurf = "lringfinger00.pss";
        attribute = flesh;
        centerofmass = (-0.10,0.95,0.08);
        mass = 15.76g;
        site base0->location = trans(-0.22cm,0.29cm,-0.11cm);
        site tip->location = trans(-0.22cm,1.29cm,-0.11cm);
site NS_proximal->location = trans(-0.22cm,0.29cm,-0.11cm);
        site NS_distal->location = trans(-0.22cm,1.29cm,-0.11cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.38,5.28,1.01),
(1.7,5.28,1.65),"y");
    }
    segment left_finger12 {
        psurf = "lmidfinger02.pss" * scale(0.99,2.84,0.92);
        attribute = flesh;
        centerofmass = (0.05,2.03,-0.01);
        mass = 3.44g;
        site base0->location = trans(0.06cm,0.77cm,-0.06cm);
        site tip->location = trans(0.07cm,3.61cm,0.36cm);
site NS_proximal->location = trans(0.06cm,0.77cm,-0.06cm);
        site NS_distal->location = trans(0.07cm,3.61cm,0.36cm);
site LM_dactylion_III_1->location = trans(0.01cm,3.37cm,0.74cm);
    }
    segment left_finger11 {
        psurf = "lmidfinger01.pss";
        attribute = flesh;
        centerofmass = (-0.06,0.64,-0.01);
        mass = 4.33g;
        site base0->location = trans(-0.05cm,0.24cm,-0.04cm);
        site tip->location = trans(-0.07cm,1.24cm,-0.03cm);
site NS_proximal->location = trans(-0.05cm,0.24cm,-0.04cm);
        site NS_distal->location = trans(-0.07cm,1.24cm,-0.03cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.12,2.62,1.11),
(1.18,2.62,1.1),"y");
    }
    segment left_finger10 {
        psurf = "lmidfinger00.pss";
```

```
        attribute = flesh;
        centerofmass = (0.05,0.83,0.04);
        mass = 18.02g;
        site base0->location = trans(0.03cm,0.26cm,0.00cm);
        site tip->location = trans(0.04cm,1.26cm,0.01cm);
site NS_proximal->location = trans(0.03cm,0.26cm,0.00cm);
        site NS_distal->location = trans(0.04cm,1.26cm,0.01cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.46,5.46,1.08),
(1.79,5.46,1.77),"y");
    }
    segment left_finger02 {
psurf = "linfinger02.pss" * scale (1,2.84,0.91);
attribute = flesh;
centerofmass = (-0.08,2.04,-0.13);
mass = 3.54g;
site base0->location = trans(-0.02cm,0.74cm,0.00cm);
site tip->location = trans(-0.10cm,3.57cm,0.47cm);
site NS_proximal->location = trans(-0.02cm,0.74cm,0.00cm);
site NS_distal->location = trans(-0.10cm,3.57cm,0.47cm);
site LM_dactylion_II->location = trans(-0.16cm,3.35cm,0.83cm);
    }
    segment left_finger01 {
        psurf = "linfinger01.pss";
        attribute = flesh;
        centerofmass = (0.00,0.62,-0.08);
        mass = 3.86g;
        site base0->location = trans(0.13cm,0.27cm,0.03cm);
        site tip->location = trans(0.14cm,1.27cm,0.04cm);
site NS_proximal->location = trans(0.13cm,0.27cm,0.03cm);
        site NS_distal->location = trans(0.14cm,1.27cm,0.04cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.14,2.25,1.09),
(1.2,2.25,1.09),"y");
    }
    segment left_finger00 {
        psurf = "linfinger00.pss";
        attribute = flesh;
        centerofmass = (0.20,1.12,-0.07);
        mass = 15.65g;
        site base0->location = trans(0.34cm,0.25cm,0.17cm);
        site tip->location = trans(0.34cm,1.25cm,0.18cm);
site NS_proximal->location = trans(0.34cm,0.25cm,0.17cm);
        site NS_distal->location = trans(0.34cm,1.25cm,0.18cm);
site LM_metacarpale_II->location = trans(1.00cm,0.27cm,0.17cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(1.49,4.54,1.07),
(1.83,4.54,1.74),"y");
    }
```

```
    segment left_thumb2 {
        psurf = "lthumb02.pss" * scale(1.20,3.44,1.14);
        attribute = flesh;
        centerofmass = (-0.01,3.18,0.09);
        mass = 6.13g;
        site base0->location = trans(0.00cm,1.02cm,0.07cm);
        site tip->location = trans(-0.32cm,4.46cm,-0.73cm);
site NS_proximal->location = trans(0.00cm,1.02cm,0.07cm);
        site NS_distal->location = trans(-0.32cm,4.46cm,-0.73cm);
site LM_thumbtip->location = trans(-0.17cm,4.45cm,-0.79cm);
    }
    segment left_thumb1 {
        psurf = "lthumb01.pss";
        attribute = flesh;
        centerofmass = (0.03,0.49,-0.08);
        mass = 5.00g;
        site base0->location = trans(0.00cm,0.25cm,0.10cm);
        site tip->location = trans(0.02cm,1.25cm,0.10cm);
site NS_proximal->location = trans(0.00cm,0.25cm,0.10cm);
        site NS_distal->location = trans(0.02cm,1.25cm,0.10cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.2,2.11,1.14),
(1.55,2.11,1.48),"y");
    }
    segment left_thumb0 {
        psurf = "lthumb00.pss";
        attribute = flesh;
        centerofmass = (-0.11,0.63,-0.01);
        mass = 73.77g;
        site tip->location = trans(-0.14cm,1.26cm,0.04cm);
        site base0->location = xyz(0.00deg,0.00deg,-49.75deg) *
trans(-0.14cm,0.26cm,0.06cm);
site NS_proximal->location = trans(-0.14cm,1.26cm,0.04cm);
        site NS_distal->location = xyz(0.00deg,0.00deg,-49.75deg) *
trans(-0.14cm,0.26cm,0.06cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(2.63,8.18,1.71),
(2.75,8.18,2.28),"y");
    }
    segment right_thumb2 {
        psurf = "rthumb02.pss" * scale(1.20,3.44,1.14);
        attribute = flesh;
        centerofmass = (-0.01,3.18,0.09);
        mass = 6.13g;
        site base0->location = trans(0.00cm,1.02cm,-0.07cm);
        site tip->location = trans(-0.32cm,4.46cm,0.73cm);
site NS_proximal->location = trans(0.00cm,1.02cm,-0.07cm);
        site NS_distal->location = trans(-0.32cm,4.46cm,0.73cm);
```

```
site LM_thumbtip->location = trans(-0.17cm,4.45cm,0.79cm);
    }
    segment right_thumb1 {
        psurf = "rthumb01.pss";
        attribute = flesh;
        centerofmass = (0.03,0.49,-0.08);
        mass = 5.00g;
        site base0->location = trans(0.00cm,0.25cm,-0.10cm);
        site tip->location = trans(0.02cm,1.25cm,-0.10cm);
site NS_proximal->location = trans(0.00cm,0.25cm,-0.10cm);
site NS_distal->location = trans(0.02cm,1.25cm,-0.10cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.2,2.11,1.14),
(1.55,2.11,1.48),"y");
    }
    segment right_thumb0 {
        psurf = "rthumb00.pss";
        attribute = flesh;
        centerofmass = (-0.11,0.63,-0.01);
mass = 73.77g;
site base0->location = xyz(0.00deg,0.00deg,-49.75deg) *
trans(-0.14cm,0.26cm,-0.06cm);
site tip->location = trans(-0.14cm,1.26cm,-0.04cm);
site NS_proximal->location = xyz(0.00deg,0.00deg,-49.75deg) *
trans(-0.14cm,0.26cm,-0.06cm);
site NS_distal->location = trans(-0.14cm,1.26cm,-0.04cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(2.63,8.18,1.71),
(2.75,8.18,2.28),"y");
    }
    segment right_finger32 {
        psurf = "rpinfinger02.pss" * scale(0.87,2.72,0.78);
        attribute = flesh;
        centerofmass = (0.02,1.91,-0.04);
        mass = 2.54g;
        site base0->location = trans(0.00cm,0.68cm,0.08cm);
        site tip->location = trans(0.07cm,3.40cm,-0.44cm);
site NS_proximal->location = trans(0.00cm,0.68cm,0.08cm);
        site NS_distal->location = trans(0.07cm,3.40cm,-0.44cm);
    }
    segment right_finger31 {
        psurf = "rpinfinger01.pss";
        attribute = flesh;
        centerofmass = (-0.04,0.65,-0.01);
        mass = 2.17g;
        site base0->location = trans(-0.06cm,0.33cm,0.03cm);
        site tip->location = trans(-0.06cm,1.33cm,0.01cm);
site NS_proximal->location = trans(-0.06cm,0.33cm,0.03cm);
```

```
        site NS_distal->location = trans(-0.06cm,1.33cm,0.01cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(0.95,1.75,0.92),
(1.04,1.75,0.94),"y");
    }
    segment right_finger30 {
        psurf = "rpinfinger00.pss";
        attribute = flesh;
        centerofmass = (-0.09,0.91,0.10);
        mass = 9.94g;
        site base0->location = trans(-0.23cm,0.28cm,0.09cm);
        site tip->location = trans(-0.23cm,1.28cm,0.09cm);
site NS_proximal->location = trans(-0.23cm,0.28cm,0.09cm);
site NS_distal->location = trans(-0.23cm,1.28cm,0.09cm);
site LM_metacarpale_V->location = trans(-0.76cm,0.24cm,0.12cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(1.24,4.14,0.9),
(1.53,4.14,1.47),"y");
    }
    segment right_finger22 {
        psurf = "rringfinger02.pss" * scale(0.92,2.97,0.86);
        attribute = flesh;
        centerofmass = (0.02,2.06,-0.11);
        mass = 3.14g;
        site base0->location = trans(-0.01cm,0.60cm,-0.09cm);
        site tip->location = trans(0.04cm,3.57cm,-0.47cm);
site NS_proximal->location = trans(-0.01cm,0.60cm,-0.09cm);
site NS_distal->location = trans(0.04cm,3.57cm,-0.47cm);
    }
    segment right_finger21 {
        psurf = "rringfinger01.pss";
        attribute = flesh;
        centerofmass = (0.03,0.59,0.10);
        mass = 3.55g;
        site base0->location = trans(0.02cm,0.25cm,0.06cm);
        site tip->location = trans(0.01cm,1.25cm,0.07cm);
site NS_proximal->location = trans(0.02cm,0.25cm,0.06cm);
        site NS_distal->location = trans(0.01cm,1.25cm,0.07cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.07,2.42,1.03),
(1.1,2.42,1.03),"y");
    }
    segment right_finger20 {
        psurf = "rringfinger00.pss";
        attribute = flesh;
        centerofmass = (-0.10,0.95,0.08);
        mass = 15.76g;
        site base0->location = trans(-0.22cm,0.29cm,0.11cm);
        site tip->location = trans(-0.22cm,1.29cm,0.11cm);
```

```
site NS_proximal->location = trans(-0.22cm,0.29cm,0.11cm);
        site NS_distal->location = trans(-0.22cm,1.29cm,0.11cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.38,5.28,1.01),
(1.7,5.28,1.65),"y");
    }
    segment right_finger12 {
        psurf = "rmidfinger02.pss" * scale(0.99,2.84,0.92);
        attribute = flesh;
        centerofmass = (0.05,2.03,-0.01);
        mass = 3.44g;
        site base0->location = trans(0.06cm,0.77cm,0.06cm);
        site tip->location = trans(0.07cm,3.61cm,-0.36cm);
site NS_proximal->location = trans(0.06cm,0.77cm,0.06cm);
        site NS_distal->location = trans(0.07cm,3.61cm,-0.36cm);
site LM_dactylion_III_r->location = trans(0.00cm,3.38cm,-0.73cm);
    }
    segment right_finger11 {
        psurf = "rmidfinger01.pss";
        attribute = flesh;
        centerofmass = (-0.06,0.64,-0.01);
        mass = 4.33g;
        site base0->location = trans(-0.05cm,0.24cm,0.04cm);
        site tip->location = trans(-0.07cm,1.24cm,0.03cm);
site NS_proximal->location = trans(-0.05cm,0.24cm,0.04cm);
        site NS_distal->location = trans(-0.07cm,1.24cm,0.03cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.12,2.62,1.11),
(1.18,2.62,1.1),"y");
    }
    segment right_finger10 {
        psurf = "rmidfinger00.pss";
        attribute = flesh;
        centerofmass = (0.05,0.83,0.04);
        mass = 18.02g;
        site base0->location = trans(0.03cm,0.26cm,0.00cm);
        site tip->location = trans(0.04cm,1.26cm,-0.01cm);
site NS_proximal->location = trans(0.03cm,0.26cm,0.00cm);
        site NS_distal->location = trans(0.04cm,1.26cm,-0.01cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.46,5.46,1.08),
(1.79,5.46,1.77),"y");
    }
    segment right_finger02 {
psurf = "rinfinger02.pss" * scale(1,2.84,0.91);
attribute = flesh;
centerofmass = (-0.08,2.04,-0.13);
mass = 3.54g;
site base0->location = trans(-0.02cm,0.74cm,0.00cm);
```

```
site tip->location = trans(-0.10cm,3.57cm,-0.47cm);
site NS_proximal->location = trans(-0.02cm,0.74cm,0.00cm);
site NS_distal->location = trans(-0.10cm,3.57cm,-0.47cm);
site LM_dactylion_II->location = trans(-0.16cm,3.35cm,-0.83cm);
    }
    segment right_finger01 {
        psurf = "rinfinger01.pss";
        attribute = flesh;
        centerofmass = (0.00,0.62,-0.08);
        mass = 3.86g;
        site base0->location = trans(0.13cm,0.27cm,-0.03cm);
        site tip->location = trans(0.14cm,1.27cm,-0.04cm);
site NS_proximal->location = trans(0.13cm,0.27cm,-0.03cm);
        site NS_distal->location = trans(0.14cm,1.27cm,-0.04cm);
site_scale = ((site)NS_proximal,(site)NS_distal,(1.14,2.25,1.09),
(1.2,2.25,1.09),"y");
    }
    segment right_finger00 {
        psurf = "rinfinger00.pss";
        attribute = flesh;
        centerofmass = (0.20,1.12,-0.07);
        mass = 15.65g;
        site base0->location = trans(0.34cm,0.25cm,-0.17cm);
        site tip->location = trans(0.34cm,1.25cm,-0.18cm);
site NS_proximal->location = trans(0.34cm,0.25cm,-0.17cm);
        site NS_distal->location = trans(0.34cm,1.25cm,-0.18cm);
site LM_metacarpale_II->location = trans(1.00cm,0.27cm,-0.17cm);
        site_scale = ((site)NS_proximal,(site)NS_distal,(1.49,4.54,1.07),
(1.83,4.54,1.74),"y");
    }
    segment right_palm {
        psurf = "rpalm.pss" * scale(4.76,8.45,1.65);
        attribute = flesh;
        centerofmass = (0.55,3.72,0.07);
        mass = 313.82g;
        site NS_proximal->location = xyz(-90.00deg,-6.56deg,0.00deg) *
trans(1.62cm,0.00cm,0.02cm);
site NS_distal->location = xyz(2.98deg,0.14deg,4.87deg) *
trans(1.04cm,8.45cm,-0.06cm);
site base->location = xyz(-90.00deg,-6.56deg,0.00deg) *
trans(1.62cm,0.00cm,0.02cm);
site real_base->location = xyz(-90.00deg,-6.56deg,0.00deg) *
trans(1.62cm,0.60cm,0.02cm);
        site f11->location = xyz(2.98deg,0.14deg,4.87deg) *
trans(3.52cm,8.80cm,0.01cm);
        site f22->location = xyz(2.98deg,0.14deg,4.87deg) *
```

```
trans(1.04cm,8.45cm,-0.06cm);
site f33->location = xyz(2.98deg,0.14deg,4.87deg) *
trans(-1.58cm,8.14cm,0.20cm);
        site f44->location = xyz(2.98deg,0.14deg,4.87deg) *
trans(-3.92cm,7.32cm,0.48cm);
        site right_bird->location = xyz(-86.00deg,-6.39deg,0.12deg) *
trans(0.01cm,7.35cm,-0.69cm);
        site palmcenter->location = trans(0.19cm,5.77cm,-0.08cm);
        site thumb0->location = xyz(-177.02deg,-0.14deg,-164.87deg) *
trans(1.85cm,1.96cm,0.62cm);
        site left->location = trans(-4.75cm,7.27cm,0.44cm);
        site right->location = trans(4.76cm,7.27cm,0.44cm);
        site front->location = trans(-0.05cm,5.55cm,-1.45cm);
        site back->location = trans(-0.05cm,5.55cm,1.53cm);
        site LM_hand_closed_center->location = trans(0.19cm,7.77cm,-0.08cm);
    }
    joint rthumb1 {
        connect right_thumb0.tip to right_thumb1.base0;
        type = R(x);
        llimit = (-45.00deg);
        ulimit = (5.00deg);
    }
    joint rthumb2 {
        connect right_thumb1.tip to right_thumb2.base0;
        type = R(x);
        llimit = (-75.00deg);
        ulimit = (10.00deg);
    }
    joint rpinfinger31 {
        connect right_finger30.tip to right_finger31.base0;
        type = R(x);
        llimit = (-5.00deg);
        ulimit = (95.00deg);
    }
    joint rpinfinger32 {
        connect right_finger31.tip to right_finger32.base0;
        type = R(x);
        llimit = (0.00deg);
        ulimit = (60.00deg);
    }
    joint rringfinger22 {
        connect right_finger21.tip to right_finger22.base0;
        type = R(x);
        llimit = (0.00deg);
        ulimit = (60.00deg);
    }
```

```
joint rringfinger21 {
    connect right_finger20.tip to right_finger21.base0;
    type = R(x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint rmidfinger12 {
    connect right_finger11.tip to right_finger12.base0;
    type = R(x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint rmidfinger11 {
    connect right_finger10.tip to right_finger11.base0;
    type = R(x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint rinfinger02 {
    connect right_finger01.tip to right_finger02.base0;
    type = R(x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint rinfinger01 {
    connect right_finger00.tip to right_finger01.base0;
    type = R(x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint right_finger00 {
    connect right_palm.f11 to right_finger00.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint right_finger10 {
    connect right_palm.f22 to right_finger10.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint right_finger20 {
    connect right_palm.f33 to right_finger20.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
```

```
    ulimit = (30.00deg,80.00deg);
}
joint right_finger30 {
    connect right_palm.f44 to right_finger30.base0;
    type = R(z) * R(x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint rthumb0 {
    connect right_palm.thumb0 to right_thumb0.base0;
    type = R(-z) * R(-y);
    llimit = (0.00deg,0.00deg);
    ulimit = (40.00deg,110.00deg);
}
joint right_toes {
    connect right_foot.toes to right_toes.proximal;
    type = R(y);
    llimit = (0.00deg);
    ulimit = (90.00deg);
}
joint left_toes {
    connect left_foot.toes to left_toes.proximal;
    type = R(y);
    llimit = (0.00deg);
    ulimit = (90.00deg);
}
joint right_ankle {
    connect right_lower_leg.distal to right_foot.proximal;
    type = R(-z) * R(-x) * R(y);
    llimit = (-55.00deg,-39.00deg,-79.60deg);
    ulimit = (63.00deg,35.00deg,25.00deg);
}
joint left_ankle {
    connect left_lower_leg.distal to left_foot.proximal;
    type = R(z) * R(x) * R(y);
    llimit = (-55.00deg,-39.00deg,-79.60deg);
    ulimit = (63.00deg,35.00deg,25.00deg);
}
joint right_knee {
    connect right_upper_leg.distal to right_lower_leg.proximal;
    type = R(-y);
    llimit = (0.00deg);
    ulimit = (160.20deg);
    rest = (10.00deg);
    tolerance = (10.00deg);
}
```

```
joint left_knee {
    connect left_upper_leg.distal to left_lower_leg.proximal;
    type = R(-y);
    llimit = (0.00deg);
    ulimit = (160.20deg);
    rest = (10.00deg);
    tolerance = (10.00deg);
}
joint right_hip {
    connect lower_torso.rhip_lateral to right_upper_leg.proximal;
    type = R(-z) * R(-x) * R(y);
    llimit = (-50.00deg,-10.00deg,-17.00deg);
    ulimit = (40.00deg,30.00deg,117.00deg);
}
joint left_hip {
    connect lower_torso.lhip_lateral to left_upper_leg.proximal;
    type = R(z) * R(x) * R(y);
    llimit = (-50.00deg,-10.00deg,-17.00deg);
    ulimit = (40.00deg,30.00deg,117.00deg);
}
joint atlanto_occipital {
    connect neck.distal to bottom_head.proximal;
    type = R(z) * R(y) * R(x);
    llimit = (-43.50deg,-51.50deg,-23.50deg);
    ulimit = (43.50deg,25.80deg,23.50deg);
}
joint base_of_neck {
    connect t1.distal to neck.proximal;
    type = R(y) * R(z) * R(x);
    llimit = (-51.50deg,-56.10deg,-20.00deg);
    ulimit = (45.20deg,56.10deg,20.00deg);
}
joint spinet2t1 {
    connect t2.distal to t1.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet3t2 {
    connect t3.distal to t2.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet4t3 {
    connect t4.distal to t3.proximal;
```

```
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet5t4 {
    connect t5.distal to t4.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet6t5 {
    connect t6.distal to t5.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
joint spinet7t6 {
    connect t7.distal to t6.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-4.00deg,-1.67deg);
    ulimit = (2.50deg,4.00deg,1.67deg);
}
joint spinet8t7 {
    connect t8.distal to t7.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-2.00deg,-1.67deg);
    ulimit = (2.50deg,2.00deg,1.67deg);
}
joint spinet9t8 {
    connect t9.distal to t8.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-2.00deg,-1.67deg);
    ulimit = (2.50deg,2.00deg,1.67deg);
}
joint spinet10t9 {
    connect t10.distal to t9.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-2.00deg,-1.67deg);
    ulimit = (2.50deg,2.00deg,1.67deg);
}
joint spinet11t10 {
    connect t11.distal to t10.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-1.67deg,-3.00deg,-1.67deg);
    ulimit = (2.50deg,3.00deg,1.67deg);
}
```

```
joint spinet12t11 {
    connect t12.distal to t11.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-2.40deg,-2.00deg,-1.67deg);
    ulimit = (3.00deg,2.00deg,1.67deg);
}
joint spinel1t12 {
    connect l1.distal to t12.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-3.30deg,-5.00deg,-1.67deg);
    ulimit = (4.00deg,5.00deg,1.67deg);
}
joint spinel2l1 {
    connect l2.distal to l1.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-3.97deg,-1.00deg,-4.00deg);
    ulimit = (7.03deg,1.00deg,4.00deg);
}
joint spinel3l2 {
    connect l3.distal to l2.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-4.33deg,-1.50deg,-4.00deg);
    ulimit = (7.67deg,1.50deg,4.00deg);
}
joint spinel4l3 {
    connect l4.distal to l3.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-6.50deg,-1.50deg,-4.00deg);
    ulimit = (11.50deg,1.50deg,4.00deg);
}
joint spinel5l4 {
    connect l5.distal to l4.proximal;
    type = R(x) * R(y) * R(z);
    llimit = (-8.66deg,-2.00deg,-4.00deg);
    ulimit = (15.34deg,2.00deg,4.00deg);
}
joint waist {
    connect lower_torso.distal to l5.proximal;
    type = R(y) * R(z) * R(x);
    llimit = (-6.50deg,-2.00deg,-4.00deg);
    ulimit = (11.00deg,2.00deg,4.00deg);
}
joint solar_plexus {
    connect t1.distal to upper_torso.proximal;
    displacement = trans(0.00cm,0.00cm,0.00cm);
}
```

```
joint right_clavicle_joint {
    connect upper_torso.rclav to right_clavicle.proximal;
    type = R(-x) * R(y);
    llimit = (-12.00deg,-8.00deg);
    ulimit = (25.00deg,40.00deg);
}
joint right_shoulder {
    connect right_clavicle.lateral to right_upper_arm.proximal;
    type = R(-z) * R(-x) * R(y);
    llimit = (-108.55deg,-48.00deg,-61.45deg);
    ulimit = (71.50deg,180.95deg,187.65deg);
}
joint right_elbow {
    connect right_upper_arm.distal to right_lower_arm.proximal;
    type = R(y);
    llimit = (0.00deg);
    ulimit = (149.75deg);
    rest = (5.00deg);
    tolerance = (5.00deg);
}
joint right_wrist {
    connect right_lower_arm.distal to right_palm.base;
    type = R(y) * R(-x) * R(-z);
    llimit = (-45.00deg,-85.00deg,-90.00deg);
    ulimit = (45.00deg,100.00deg,94.80deg);
}
joint left_elbow {
    connect left_upper_arm.distal to left_lower_arm.proximal;
    type = R(y);
    llimit = (0.00deg);
    ulimit = (149.75deg);
    rest = (5.00deg);
    tolerance = (5.00deg);
}
joint left_wrist {
    connect left_lower_arm.distal to left_palm.base;
    type = R(y) * R(x) * R(z);
    llimit = (-45.00deg,-85.00deg,-90.00deg);
    ulimit = (45.00deg,100.00deg,94.80deg);
}
joint left_clavicle_joint {
    connect upper_torso.lclav to left_clavicle.proximal;
    type = R(x) * R(y);
    llimit = (-12.00deg,-8.00deg);
    ulimit = (25.00deg,40.00deg);
}
```

```
joint left_shoulder {
    connect left_clavicle.lateral to left_upper_arm.proximal;
    type = R(z) * R(x) * R(y);
    llimit = (-108.55deg,-48.00deg,-61.45deg);
    ulimit = (71.50deg,180.95deg,187.65deg);
}
joint linfinger02 {
    connect left_finger01.tip to left_finger02.base0;
    type = R(-x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint linfinger01 {
    connect left_finger00.tip to left_finger01.base0;
    type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint lpinfinger31 {
    connect left_finger30.tip to left_finger31.base0;
    type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint lpinfinger32 {
    connect left_finger31.tip to left_finger32.base0;
    type = R(-x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint lringfinger22 {
    connect left_finger21.tip to left_finger22.base0;
    type = R(-x);
    llimit = (0.00deg);
    ulimit = (60.00deg);
}
joint lringfinger21 {
    connect left_finger20.tip to left_finger21.base0;
    type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint lmidfinger12 {
    connect left_finger11.tip to left_finger12.base0;
    type = R(-x);
    llimit = (0.00deg);
```

```
    ulimit = (60.00deg);
}
joint lmidfinger11 {
    connect left_finger10.tip to left_finger11.base0;
    type = R(-x);
    llimit = (-5.00deg);
    ulimit = (95.00deg);
}
joint left_finger00 {
    connect left_palm.f11 to left_finger00.base0;
    type = R(z) * R(-x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint left_finger10 {
    connect left_palm.f22 to left_finger10.base0;
    type = R(z) * R(-x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint left_finger20 {
    connect left_palm.f33 to left_finger20.base0;
    type = R(z) * R(-x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint left_finger30 {
    connect left_palm.f44 to left_finger30.base0;
    type = R(z) * R(-x);
    llimit = (-30.00deg,-10.00deg);
    ulimit = (30.00deg,80.00deg);
}
joint lthumb2 {
    connect left_thumb1.tip to left_thumb2.base0;
    type = R(-x);
    llimit = (-75.00deg);
    ulimit = (10.00deg);
}
joint lthumb1 {
    connect left_thumb0.tip to left_thumb1.base0;
    type = R(-x);
    llimit = (-45.00deg);
    ulimit = (5.00deg);
}
joint lthumb0 {
    connect left_palm.thumb0 to left_thumb0.base0;
```

```
        type = R(-z) * R(y);
        llimit = (0.00deg,0.00deg);
        ulimit = (40.00deg,110.00deg);
    }
    joint right_eyeball {
        connect bottom_head.right_eyeball to right_eyeball.base;
        type = R(x) * R(z);
        llimit = (-20.00deg,-30.00deg);
        ulimit = (20.00deg,30.00deg);
    }
    joint left_eyeball {
        connect bottom_head.left_eyeball to left_eyeball.base;
        type = R(x) * R(z);
        llimit = (-20.00deg,-30.00deg);
        ulimit = (20.00deg,30.00deg);
    }

    posturedef ["mysit.post"] mysit;
    posturedef ["mystand.post"] mystand;
    posturedef ["functionalzero.post"] functionalzero;
    posturedef ["functionalone.post"] functionalone;
    posturedef ["overhead.post"] overhead;
    posturedef ["sitting_forearm_up.post"] sitting_forearm_up;
    posturedef ["forearm_up.post"] forearm_up;
    posturedef ["forearm_up_hand_closed.post"] forearm_up_hand_closed;
    posturedef ["arm_span.post"] arm_span;
    posturedef ["biceps_flex.post"] biceps_flex;

    root = left_toes.distal;
}
```

Appendix D

Postures for Measurement

The following figures show all the postures used to perform anthropometric measurements. All these postures follow those described in ANSUR-88. Refer to chapter 5 for details on which measurements are done under the different postures.

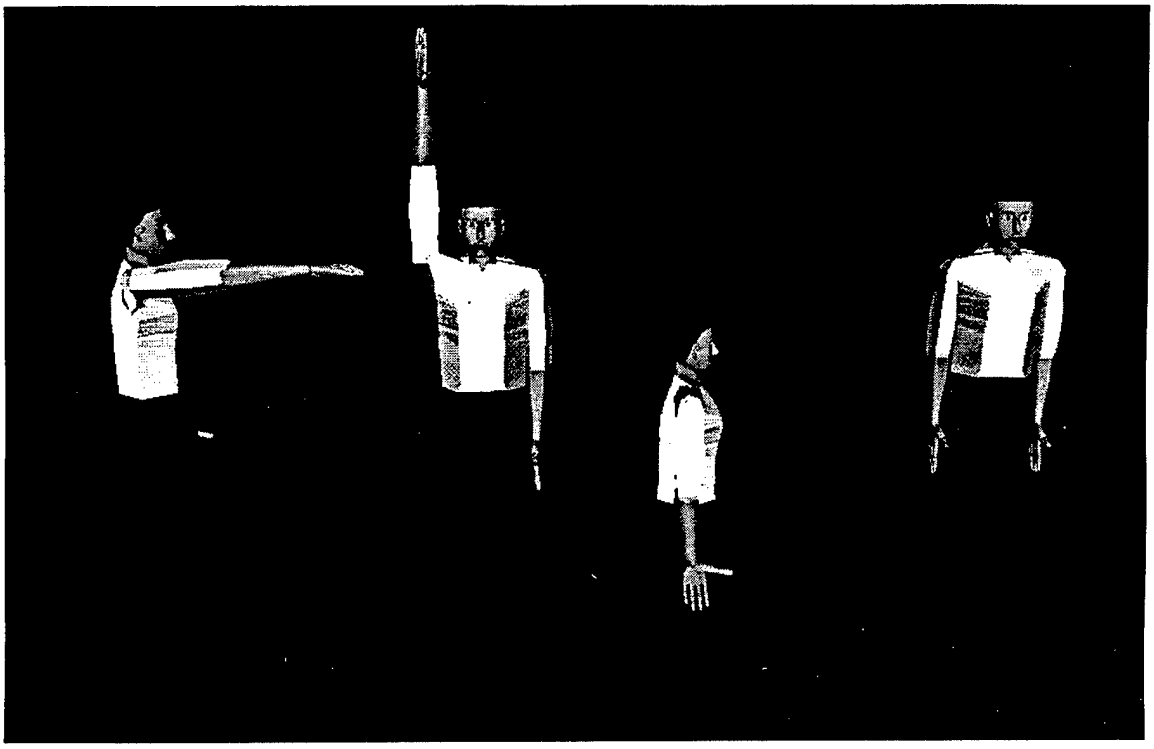

Figure D.1: From left to right: Functional, Overhead, Sitting, and Standing postures

Figure D.2: Functional-Extended posture

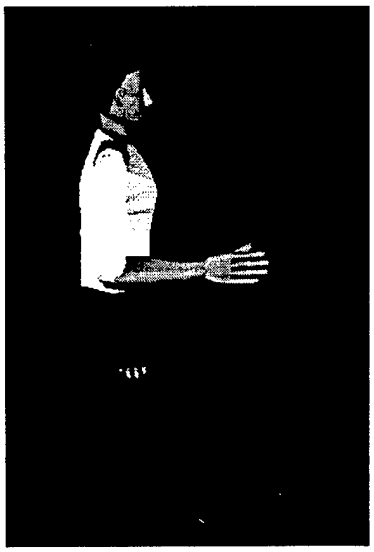

Figure D.3: Sitting Forearm Up posture

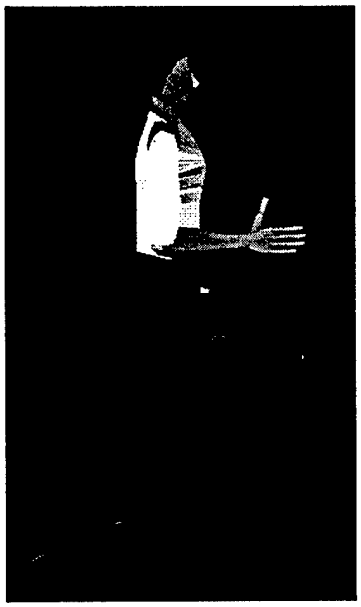

Figure D.4: Standing Forearm Up posture

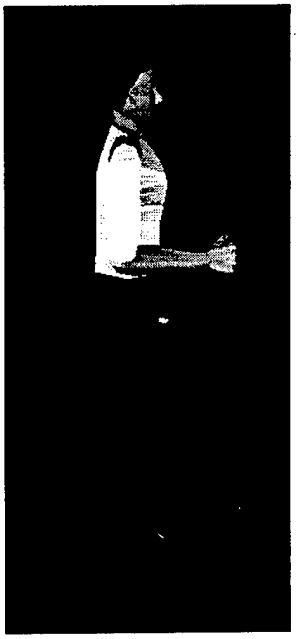

Figure D.5: Forearm Up (Hand Closed) posture

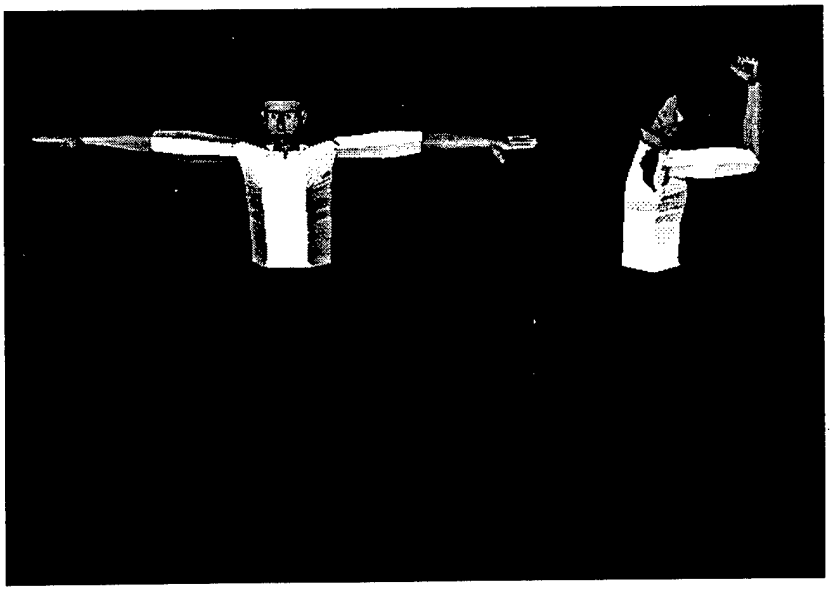

Figure D.6: From left to right: Arm span and Biceps-Flex postures

Appendix E

Torso Distribution Factors

We present the torso distribution factors used for figure creation. This factors have already been incorporated in the formulae for the target anthropometric data representation appearing in Chapter 5.

```
//
//
// Torso: 17 segments distribution
//
//                 width  length  depth
// ---------------------------------
thorax1    t1     0.8166 0.04309 0.7150
thorax2    t2     0.9477 0.04567 0.8600
thorax3    t3     0.9812 0.04235 0.9178
thorax4    t4     1.0000 0.06077 0.8900
thorax5    t5     0.9693 0.04899 0.9612
thorax6    t6     0.9693 0.05193 0.9677
thorax7    t7     0.9693 0.05193 0.9852
thorax8    t8     0.9693 0.05193 0.9852
thorax9    t9     0.9693 0.05488 0.9900
thorax10 t10    0.9693 0.05967 0.9825
thorax11 t11    0.9442 0.03867 0.9600
thorax12 t12    0.9442 0.08361 0.9350
lumbar1  l1     0.9442 0.07845 0.9000
lumbar2  l2     0.9275 0.07845 0.9000
lumbar3  l3     0.9066 0.05782 0.9000
lumbar4  l4     0.9066 0.07514 0.9000
lumbar5  l5     0.9066 0.07661 0.9300
```

Appendix F

Default Anthropometric Variables

The following table presents a summary of the default anthropometric variables considered by *Genfig* and *Jack*.

List #	Variable Name	Variable #
1	ACRHGHT	2
2	ACRHTST	3
3	ACRDLG	4
4	ANKLCIRC	5
5	BLFTLG	9
6	BCRMBDTH	10
7	BICIRCFL	11
8	BIDLBDTH	12
9	BIMBDTH	13
10	BISBDTH	14
11	BIZBDTH	21
12	BUTTDPTH	24
13	BUTTKLTH	26
14	BUTTPLTH	27
15	CALFCIRC	28
16	CERVHT	30
17	CERVSIT	31
18	CHSTBDTH	32
19	CHSTDPTH	36
20	CRCHHGHT	38
21	ELRHGHT	48
22	EYEHTSIT	49
23	FTBRHOR	50
24	FOOTLG	51
25	FCIRCFL	52
26	FORHDLG	54
27	HANDBRTH	57
28	HANDLG	59
29	HEADBRTH	60
30	HEADLGTH	62
31	HIPBRTH	65
32	ILCRSIT	67
33	INPUPBTH	68
34	INSCYE1	69

Figure F.1: Default Set of Anthropometric Variables

List #	Variable Name	Variable #
35	KNEECIRC	71
36	KNEEHTPMP	72
37	KNEEHTSIT	73
38	LATFEMEP	74
39	LATMALHT	75
40	NECKCIRC	80
41	OVHDFTRH	83
42	POPHGHT	86
43	RASTL	87
44	SHOUELLT	91
45	SHOULGTH	92
46	SITTHGT	93
47	SPAN	98
48	STATURE	99
49	SUPSTRHT	101
50	TENRIBHT	102
51	THGHCIRC	103
52	THGHCLR	104
53	THMBTPR	106
54	TROCHHT	107
55	WSTBRTH	112
56	WSTDPTH	115
57	WSTHNI	118
58	WSTHSTNI	120
59	WSHTSTOM	121
60	WEIGHT	124
61	WRCTRGRL	125
62	WRISCIRC	126
63	WRINFNGL	129
64	WRTHLGTH	130
65	WRWALLLN	131
66	WRWALLEX	132
67	ECTORBT (H15)	232
68	INFORBB (H22)	239
69	TRAGT (H44)	254

Figure F.2: Default Set of Anthropometric Variables (continued)

Appendix G

Landmarks

This is the list of landmarks currently available in the human model. Since landmark locations are not available from anthropometric surveys they are place by educated guesses.

```
Global Location of the figure's root  = (0.000000,0.000002,0.000001)

segment bottom_head

Global Location of Segment's Origin = (-9.048829,152.059723,-7.979547)

landmark.bottom_head.LM_alare_l
global=(-7.377992,159.521774,0.145453) local=(8.380000,1.690000,7.170000)
landmark.bottom_head.LM_alare_r
global=(-10.757982,159.522507,0.136798) local=(8.380000,-1.690000,7.170000)
landmark.bottom_head.LM_cheilion_l
global=(-5.685760,156.569641,-0.976336) local=(7.150000,3.380000,4.260000)
landmark.bottom_head.LM_cheilion_r
global=(-12.305872,156.151367,-0.978410) local=(7.150000,-3.240000,3.840000)
landmark.bottom_head.LM_chin
global=(-8.998285,153.443954,-0.263590) local=(7.760000,0.070000,1.110000)
landmark.bottom_head.LM_crinion
global=(-9.043282,169.232651,-0.863508) local=(7.720000,0.020000,16.910000)
landmark.bottom_head.LM_ear_bottom
global=(-16.409626,159.249008,-7.062479) local=(1.190000,-7.360000,7.150000)
landmark.bottom_head.LM_ear_pt
global=(-17.775017,162.939133,-8.547649) local=(-0.160000,-8.730000,10.890000)
landmark.bottom_head.LM_ear_top
global=(-17.259325,164.584641,-6.723486) local=(1.720000,-8.210000,12.470000)
landmark.bottom_head.LM_ectocanthus
global=(-13.732141,163.456879,-1.811425) local=(6.580000,-4.670000,11.170000)
landmark.bottom_head.LM_ectoorbitale_lft
global=(-2.707700,164.235489,-3.461925) local=(4.930000,6.350000,12.010000)
landmark.bottom_head.LM_ectoorbitale_rt
```

```
global=(-15.407663,164.238266,-3.494449) local=(4.930000,-6.350000,12.010000)
landmark.bottom_head.LM_frontotemporale_l
global=(-2.879668,165.326736,-2.600483) local=(5.830000,6.180000,13.070000)
landmark.bottom_head.LM_frontotemporale_r
global=(-15.089632,165.329422,-2.631753) local=(5.830000,-6.030000,13.070000)
landmark.bottom_head.LM_glabella
global=(-9.006763,166.189636,0.235099) local=(8.710000,0.060000,13.830000)
landmark.bottom_head.LM_gonion_lft
global=(-3.259259,154.854431,-7.563533) local=(0.500000,5.790000,2.780000)
landmark.bottom_head.LM_gonion_rt
global=(-14.839226,154.856964,-7.593189) local=(0.500000,-5.790000,2.780000)
landmark.bottom_head.LM_head_top
global=(-8.953569,175.111465,-8.056120) local=(0.740000,0.090000,23.040001)
landmark.bottom_head.LM_infraorbitale_l
global=(-6.160835,161.208618,-2.502881) local=(5.790000,2.900000,8.950000)
landmark.bottom_head.LM_infraorbitale_r
global=(-11.960818,161.209900,-2.517735) local=(5.790000,-2.900000,8.950000)
landmark.bottom_head.LM_menton
global=(-8.944624,152.279022,-1.793143) local=(6.190000,0.120000,0.000000)
landmark.bottom_head.LM_otobasion_sup
global=(-15.982209,163.560471,-5.683281) local=(2.720000,-6.930000,11.410000)
landmark.bottom_head.LM_promenton
global=(-8.998311,153.444321,-0.253596) local=(7.770000,0.070000,1.110000)
landmark.bottom_head.LM_pronasale
global=(-9.012972,160.143372,2.140517) local=(10.400000,0.060000,7.720000)
landmark.bottom_head.LM_sellion
global=(-9.006626,163.246948,-0.070829) local=(8.300000,0.060000,10.900000)
landmark.bottom_head.LM_stomion
global=(-9.008897,156.262756,0.216811) local=(8.340000,0.060000,3.910000)
landmark.bottom_head.LM_subnasale
global=(-9.018757,158.529144,0.356571) local=(8.560000,0.050000,6.170000)
landmark.bottom_head.LM_top_of_head
global=(-8.910146,175.015381,-13.305922) local=(-4.510000,0.120000,23.129999)
landmark.bottom_head.LM_tragion_lft
global=(-1.600987,162.184769,-6.258206) local=(2.060000,7.450000,10.060000)
landmark.bottom_head.LM_tragion_rt
global=(-16.500944,162.188034,-6.296365) local=(2.060000,-7.450000,10.060000)
landmark.bottom_head.LM_zygion_lft
global=(-3.159506,162.213150,-2.931087) local=(5.390000,5.900000,9.970000)
landmark.bottom_head.LM_zygion_rt
global=(-14.959473,162.215744,-2.961307) local=(5.390000,-5.900000,9.970000)
landmark.bottom_head.LM_zygofrontale_lft
global=(-3.930634,166.002853,-2.166848) local=(6.290000,5.130000,13.730000)
landmark.bottom_head.LM_zygofrontale_rt
global=(-14.190604,166.005112,-2.193123) local=(6.290000,-5.130000,13.730000)
```

```
segment 13

Global Location of Segment's Origin = (-9.005613,112.955307,-8.957743)

landmark.13.LM_tenth_rib
global=(-18.551765,113.442459,-0.858032) local=(-9.540000,0.160000,8.120000)



segment 14

Global Location of Segment's Origin = (-9.088500,109.501999,-8.871319)

landmark.14.LM_waist_ant__naval_
global=(-9.051386,111.855400,1.663213) local=(0.000000,2.090000,10.590000)
landmark.14.LM_waist_lft
global=(6.427049,111.221397,-8.895446) local=(15.470000,2.090000,0.000000)
landmark.14.LM_waist_post
global=(-9.086878,111.353027,-18.490587) local=(-0.060000,2.090000,-9.570000)
landmark.14.LM_waist_rt
global=(-24.504089,111.960106,-8.951501) local=(-15.470000,2.090000,0.000000)



segment 15

Global Location of Segment's Origin = (-9.205016,105.851471,-8.522093)

landmark.15.LM_iliocristale
global=(-23.001032,106.772087,-5.095790) local=(-13.810000,0.400000,3.470000)
landmark.15.LM_waist_ant
global=(-9.096566,109.940613,1.359975) local=(0.010000,3.740000,10.020000)
landmark.15.LM_waist_lft
global=(6.147124,109.134293,-8.993267) local=(15.250000,3.740000,-0.370000)
landmark.15.LM_waist_rt
global=(-24.330116,110.014496,-9.053874) local=(-15.240000,3.740000,-0.370000)



segment left_clavicle

Global Location of Segment's Origin = (-8.179223,141.038269,-12.673971)

landmark.left_clavicle.LM_clavicle_pt_1
```

```
global=(10.592699,147.613678,-8.927941) local=(2.990000,-6.930000,18.780001)
landmark.left_clavicle.LM_clavicle_pt_lft
global=(8.543008,149.079590,-12.831386) local=(-1.040000,-7.980000,16.719999)
landmark.left_clavicle.LM_midscye_lft
global=(8.284431,141.542007,-17.937860) local=(-5.330000,0.050000,16.450001)
landmark.left_clavicle.LM_post_scye_lft
global=(2.668499,134.764145,-16.211246) local=(-2.890000,6.610000,10.840000)



segment left_finger12

Global Location of Segment's Origin = (11.509692,71.208534,-15.926214)

landmark.left_finger12.LM_dactylion_III_1
global=(12.423116,67.881714,-15.875632) local=(0.010000,3.370000,0.740000)



segment left_foot

Global Location of Segment's Origin = (3.311255,4.943436,-20.414867)

landmark.left_foot.LM_heel_pt_lft_med
global=(-1.829579,0.039345,-18.879827) local=(1.560000,5.030000,5.010000)



segment left_upper_arm

Global Location of Segment's Origin = (10.451850,147.819794,-12.718535)

landmark.left_upper_arm.LM_acromion_1
global=(10.968409,147.874466,-13.238456) local=(-0.524112,-0.515213,0.000000)
landmark.left_upper_arm.LM_deltoid_pt_lft
global=(13.061237,138.808868,-8.492590) local=(5.137892,-2.622178,8.520000)



segment lower_torso

Global Location of Segment's Origin = (-10.055210,80.758568,-7.939765)

landmark.lower_torso.LM_ant_sup_iliac_spine_lft
global=(2.064229,96.984665,1.969514) local=(10.590000,11.580000,16.190001)
landmark.lower_torso.LM_ant_sup_iliac_spine_rt
```

```
global=(-21.202429,97.797142,2.317561) local=(10.990000,-11.700000,16.190001)
landmark.lower_torso.LM_buttock_pt_lft_lat
global=(5.738100,91.488274,-11.333358) local=(-2.940000,15.410000,11.400000)
landmark.lower_torso.LM_buttock_pt_post
global=(-9.578417,93.918655,-20.107567) local=(-11.590000,0.000000,13.680000)
landmark.lower_torso.LM_buttock_pt_rt_lat
global=(-25.063843,92.541100,-11.401647) local=(-2.940000,-15.410000,11.400000)
landmark.lower_torso.LM_gluteal_furrow_pt2
global=(-18.908493,83.789276,-16.234585) local=(-8.150000,-8.970000,3.080000)
landmark.lower_torso.LM_post_iliac_spine
global=(-12.733405,102.325615,-19.961895) local=(-11.080000,-3.440000,21.959999)
landmark.lower_torso.LM_trochanter
global=(-26.013121,91.313606,-8.336633) local=(0.070000,-16.309999,10.010000)
landmark.lower_torso.LM_trochanterion
global=(-26.732830,88.861969,-5.839246) local=(2.460000,-16.940001,7.430000)


segment neck

Global Location of Segment's Origin = (-8.715386,143.008820,-10.099344)

landmark.neck.LM_Infrathyroid
global=(-8.728626,150.549652,-4.282890) local=(6.080000,0.000000,7.330000)
landmark.neck.LM_cervicale
global=(-8.636642,154.521683,-16.431067) local=(-5.920000,0.060000,11.730000)
landmark.neck.LM_cervicale0
global=(-8.706629,146.722916,-17.105347) local=(-6.870000,-0.010000,3.960000)
landmark.neck.LM_neck_lft_lat
global=(-3.727130,147.737350,-5.101000) local=(5.150000,5.000000,4.550000)
landmark.neck.LM_neck_rt_lat
global=(-13.727101,147.739548,-5.126610) local=(5.150000,-5.000000,4.550000)
landmark.neck.LM_submandibular
global=(-8.983592,152.316559,-6.097291) local=(4.330000,-0.260000,9.160000)
landmark.neck.LM_trapezius_pt_lft
global=(-3.387363,152.310547,-12.426798) local=(-2.010000,5.320000,9.380000)
landmark.neck.LM_trapezius_pt_rt
global=(-14.027332,152.312897,-12.454046) local=(-2.010000,-5.320000,9.380000)


segment right_clavicle

Global Location of Segment's Origin = (-9.319221,141.038498,-12.676890)

landmark.right_clavicle.LM_clavicle_pt_r
```

```
global=(-28.137142,147.624252,-9.047018) local=(2.970000,6.930000,18.809999)
landmark.right_clavicle.LM_clavicle_pt_rt
global=(-26.036970,149.084030,-12.890108) local=(-1.010000,7.980000,16.719999)
landmark.right_clavicle.LM_midscye_rt
global=(-25.755480,141.648926,-18.014584) local=(-5.330000,0.050000,16.450001)
landmark.right_clavicle.LM_midshoulder
global=(-20.427019,150.267334,-12.761308) local=(-1.020000,9.170000,11.110000)
landmark.right_clavicle.LM_post_scye_rt
global=(-20.151440,134.769135,-16.269688) local=(-2.890000,-6.610000,10.840000)



segment right_finger00

Global Location of Segment's Origin = (-28.364407,79.541641,-14.074752)

landmark.right_finger00.LM_metacarpale_II
global=(-28.485756,78.326057,-12.563277) local=(1.496800,1.225800,-0.184178)



segment right_finger02

Global Location of Segment's Origin = (-28.316502,72.356384,-13.326163)

landmark.right_finger02.LM_dactylion_II
global=(-28.969954,68.965134,-13.445720) local=(-0.160000,3.351200,-0.828100)



segment right_finger12

Global Location of Segment's Origin = (-28.261786,71.207993,-16.027599)

landmark.right_finger12.LM_dactylion_III_r
global=(-28.815868,67.794975,-15.986826) local=(0.000000,3.380000,-0.730000)



segment right_finger30

Global Location of Segment's Origin = (-28.139170,80.313751,-20.830462)

landmark.right_finger30.LM_metacarpale_V
global=(-27.982082,79.315765,-21.752008) local=(-0.933584,0.993600,0.105264)
```

```
segment right_foot

Global Location of Segment's Origin = (-19.539377,5.287267,-20.619473)

landmark.right_foot.LM_fifth_metatarsophalangeal_protrusion
global=(-24.962395,2.223139,-1.533533) local=(19.100000,5.030000,3.600000)
landmark.right_foot.LM_first_metatarsophalangeal_protrusion
global=(-15.127543,3.446063,-1.077419) local=(19.450001,-4.750000,1.970000)
landmark.right_foot.LM_heel_pt_rt_lat
global=(-19.683920,0.200716,-19.845764) local=(0.690000,-0.080000,5.100000)
landmark.right_foot.LM_heel_pt_rt_med
global=(-14.749946,0.007158,-18.908821) local=(1.560000,-5.030000,5.100000)
landmark.right_foot.LM_pternion
global=(-16.325338,2.720142,-20.535004) local=(0.000000,-3.320000,2.430000)



segment right_lower_arm

Global Location of Segment's Origin = (-28.102646,117.453552,-17.217857)

landmark.right_lower_arm.LM_elbow_crease
global=(-28.115301,109.562454,-12.953102) local=(4.401768,0.000000,7.815500)
landmark.right_lower_arm.LM_stylion
global=(-27.548088,86.411102,-14.631917) local=(3.125594,-0.568000,30.992500)



segment right_lower_leg

Global Location of Segment's Origin = (-18.138632,56.291206,-5.633486)

landmark.right_lower_leg.LM_calf
global=(-23.894167,39.621296,-10.050694) local=(-0.790000,6.010000,17.139999)
landmark.right_lower_leg.LM_dorsal_juncture
global=(-18.138632,56.291206,-5.633486) local=(0.000000,0.000000,0.000000)
landmark.right_lower_leg.LM_knee_pt_ant
global=(-18.236721,50.994797,-0.524657) local=(6.110000,0.140000,4.100000)
landmark.right_lower_leg.LM_lateral_malleolus
global=(-19.608755,6.714833,-15.664397) local=(0.640000,2.210000,50.549999)
landmark.right_lower_leg.LM_medial_malleolus
global=(-15.478425,6.726800,-15.669661) local=(0.620000,-1.920000,50.599998)
landmark.right_lower_leg.LM_midpatella
global=(-17.982853,48.559208,-0.598526) local=(6.550000,-0.080000,6.500000)
```

```
landmark.right_lower_leg.LM_suprapatella
global=(-18.211292,49.708824,-0.157180) local=(6.740000,0.130000,5.280000)



segment right_thumb2

Global Location of Segment's Origin = (-27.755793,81.935722,-7.147518)

landmark.right_thumb2.LM_thumbtip_
global=(-28.475712,80.567024,-2.897352) local=(-0.170000,4.450000,0.790000)



segment right_toes

Global Location of Segment's Origin = (-20.143847,2.100529,-1.459460)

landmark.right_toes.LM_acropodion
global=(-20.540081,2.426763,5.881610) local=(7.350000,0.310000,-0.190000)



segment right_upper_arm

Global Location of Segment's Origin = (-27.946840,147.828217,-12.816874)

landmark.right_upper_arm.LM_acromion_r
global=(-28.460705,147.883102,-13.339434) local=(-0.524112,0.515213,0.000000)
landmark.right_upper_arm.LM_ant_scye_upper_arm
global=(-25.785700,136.179886,-9.580854) local=(4.430202,-2.171978,11.246401)
landmark.right_upper_arm.LM_biceps_pt
global=(-31.513697,125.747787,-8.139332) local=(6.968899,3.550011,21.470402)
landmark.right_upper_arm.LM_deltoid_pt_rt
global=(-30.581797,138.818451,-8.604359) local=(5.137892,2.622178,8.520000)
landmark.right_upper_arm.LM_radiale
global=(-33.775520,113.961807,-16.638086) local=(-0.245600,5.831000,34.080002)



segment right_upper_leg

Global Location of Segment's Origin = (-17.989824,93.975716,-7.221995)

landmark.right_upper_leg.LM_dorsal_junct
global=(-18.320169,61.901680,-17.174091) local=(-9.720000,1.040000,32.130001)
```

```
landmark.right_upper_leg.LM_dorsal_juncture
global=(-18.146263,56.930321,-13.788193) local=(-6.300000,0.940000,37.080002)
landmark.right_upper_leg.LM_lat_femoral_epicondyle__standing
global=(-22.336010,50.980331,-8.750582) local=(-1.190000,5.210000,42.910000)



segment t1

Global Location of Segment's Origin = (-8.760646,147.299805,-7.678009)

landmark.t1.LM_neck_ant
global=(-8.732594,147.216461,-3.020071) local=(0.040000,1.930000,4.240000)
landmark.t1.LM_suprasternale
global=(-8.771486,147.499634,-3.428000) local=(0.000000,2.010000,3.750000)



segment t8

Global Location of Segment's Origin = (-8.784616,133.208450,-8.145068)

landmark.t8.LM_ant_scye_torso
global=(-25.314480,135.896225,-7.724455) local=(-16.540001,2.640000,0.310000)
landmark.t8.LM_midspine
global=(-8.760368,135.234741,-19.636961) local=(0.000000,1.440000,-11.580000)
landmark.t8.LM_midspine0
global=(-8.766079,133.805603,-19.689629) local=(0.000000,0.010000,-11.560000)

Number of segments: 24
Number of LM sites: 106
```

Bibliography

[1] Tarek Alameldin. *Three Dimensional Workspace Visualization for Redundant Articulated Chains.* PhD thesis, University of Pennsylvania, 1991.

[2] James E. Anderson. *Grant's Atlas of Anatomy.* Williams and Wilkins, Baltimore, 8 edition, 1983.

[3] Webb Associates. Anthropometric Source Book, volume 2: A Handbook of Anthropometric Data. Technical Report NASA RP-1024, NASA, July 1978.

[4] Francisco Azuola. *SASS v.2.5 User's Manual.* Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, 1995.

[5] Francisco Azuola. *Anthropometric Error in Computer-Based Human Figure Models.* PhD thesis, University of Pennsylvania, 1996.

[6] Norman I. Badler, Cary B. Phillips, and Bonnie Lynn Webber. *Simulating Humans, Computer Graphics Animation and Control.* Oxford University Press, New York, 1993.

[7] Donald A. Bailey, Robert M. Malina, and Roy L. Rasmussen. The influence of exercise, physical activity, and athletic performance on the dynamics of human growth. In Frank Falkner and J.M. Tanner, editors, *Human Growth, Volume 2.* Plenum Press, 1978.

[8] John V. Basmajian. *Primary Anatomy.* Williams and Wilkins Company, Baltimore, eighth edition, 1982.

[9] N. Berme, J.P. Paul, and W.K. Purves. A biomechanical analysis of the metacarpophalangeal joint. *Journal of Biomechanics*, 10:409–412, 1977.

[10] A.C. Bittner and W.F. Moroney. The Accomodated Proportion of a Potential User Population: Compilation and Comparisons of Methods for Estimation. In *Proceedings of Human Factors Society 1974*, pages 376–381, 1974.

[11] Alvah C. Bittner. Computerized Accomodated Percentage Evaluation: Review and Prospectus. In *Proceedings of 6th Congress of the International Ergonomics Association*, pages 11–16, June 1976.

[12] Alvah C. Bittner, Floyd A. Glenn III, Regina Harris, Helene P. Iavecchia, and Robert J. Wherry. Cadre: A Family of Manikins for Workstation Design. In *Trends in Ergonomics/Human Factors IV*, pages 733–740. North Holland, 1987.

[13] G.E.P. Box and Mervin E. Muller. A Note on the Generation of Random Normal Deviates. *Annals Math. Stat.*, 29:610–611, 1958.

[14] Paul W. Brand. *Clinical Mechanics of the Hand.* Mosby, St. Louis, 1985.

[15] Richard Bronson. *Schaum's Outline of Theory and Problems of Matrix Operations.* McGraw-Hill Book, Inc, New York, 1989.

[16] John E. Chadwick, David R. Haumann, and Richard E. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3), 1989.

[17] Don B. Chaffin and Gunnar B.J. Andersson. *Occupational Biomechanics.* John Wiley and Sons, Inc., New York, 2 edition, 1991.

[18] David T. Chen and David Zeltzer. Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 89–98, July 1992.

[19] James Cheverud, Claire C. Gordon, Robert A. Walker, Cashell Jacquish, Luci Kohn, Allen Moore, and Nyuta Yamashita. 1988 Anthropometry Survey of U.S. Army Personnel: Correlation Coefficients and Regression Equations (Male). Technical Report TR-90-033, United States Army NATICK, Natick, Massachusetts, 1990.

[20] W.P. Cooney, M.J. Lucca, E.Y.S. Chao, and R.L. Linsheid. The kinesiology of the thumb trapeziometacarpal joint. *Journal of Bone and Joint Surgery*, (63A):1371–1381, 1981.

[21] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, San Diego, CA, 1990.

[22] Claire C. Gordon, Bruce Bradtmiller, Thomas Churchill, Charles E. Clauser, John T. McConville, Ilse Tebbetts, and Robert A. Walker. 1988 anthropometric survey of u.s. army personnel: Methods and summary statistics. Technical Report Natick/TR-89/044, U.S. Army Natick Research, Development and Engineering Center, September 1989.

[23] Claire C. Gordon, Bruce Bradtmiller, Thomas Churchill, Charles E. Clauser, John T. McConville, Ilse Tebbetts, and Robert A. Walker. 1988 Anthropometry Survey of U.S. Army Personnel: Methods and Summary Statistics. Technical Report Natick/TR-89/044, U.S. Army Natick Research, Development and Engineering Center, September 1989.

[24] Thomas M. Greiner. Hand Anthropometry of U.S. Army Personnel. Technical Report TR-92-011, United States Army Natick, Natick, Massachusetts, December 1991.

[25] Marc R. Grosso, Richard D. Quach, Ernest Otani, Jianmin Zhao, Susanna Wei, Pei-Hwa Ho, Jiahe Lu, and Norman I. Badler. Anthropometry for Computer Human Figures. Technical Report MS-CIS-89-71, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa 19104, November 1989.

[26] Marc R. Grosso, Richard D. Quach, Ernest Otani, Jianmin Zhao, Susanna Wei, Pei-Hwa Ho, Jiahe Lu, and Norman I. Badler. Anthropometry for computer graphics human figures. Technical Report MS-CIS-89-71, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389, November 1989.

[27] F. C. T. Van Der Helm, H. E. J. Veeger, G. M. Pronk, L. H. V. Van Der Woude, and R. H. Rozendal. Geometry parameters for musculoskeletal modelling of the shoulder system. *Journal of Biomechanics*, 25(2):129–144, 1991.

[28] Jr. John A. Roebuck. *Anthropometric Methods: Designing to Fit the Human Body.* Human Factors and Ergonomics Society, Santa Monica, California, 1995.

[29] J. Joseph. Further studies of the metacarpo-phalangeal and interphalangeal joints of the thumb. *Journal of Anatomy*, (85):221–229, 1951.

[30] R. Louis. *Surgery of the Spine.* Springer-Verlag, Berlin, 1983.

[31] R.M.H. McMinn and R.T. Hutchings. *A Colour Atlas of Human Anatomy.* Wolfe Medical Publications, Ltd., London, 1977.

[32] R.S. Meindl, G.F. Zehner, and J.A. Hudson. A multivariate anthropometric method for crew station design (statistical techniques). Technical Report AL:CF-TR-1993-0054, Aerospace Medical Research Lab, Wright-Patterson Air Force Base, OH, 1993.

[33] Nicholas Metropolis and S. Ulam. The Monte Carlo Method. *Journal of The American Statistical Association*, pages 335–341, 1949.

[34] Chairman Michael Biferno. Human modeling technology and standards. October 1995.

[35] Gary Monheit and Norman I. Badler. A kinematic model of the human spine and torso. *IEEE Computer Graphics and Applications*, 11(2), 1991.

[36] Donald F. Morrison. *Multivariate Statistical Methods.* McGraw-Hill Book Co., New York, 1967.

[37] Mervin E. Muller. A Comparison of Methods for Generating Normal Deviates on Digitial Computers. *Journal of the ACM*, 6:376–383, 1959.

[38] National Aeronatic and Space Administration. *Man Systems Integration Standards*, 1987.

[39] Ernest Otani. Software tools for dynamic and kinematic modeling of human motion. Technical Report MS-CIS-89-43, Department of Computer and Information Science, University of Pennsylvania, 1989.

[40] Scott R. Parry. *Free-Form Deformation in a constructive solid geometry modeling system.* PhD thesis, The Brygham Young University, 1986.

[41] Richard P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, MA, 1981.

[42] Cary B. Phillips. *Interactive Postural Control of Articulated Geometric Figures.* PhD thesis, University of Pennsylvania, 1991.

[43] Cary B. Phillips. Jack 5 user's guide. Technical Report MS-CIS-91-78, Department of Computer and Information Science, University of Pennsylvania, 1991.

[44] Cary B. Phillips. Programming with jack. Technical Report MS-CIS-91-19, Department of Computer and Information Science, University of Pennsylvania, 1991.

[45] Les Piegl. On nurbs: A survey. *IEEE Computer Graphics and Applications,* 11(1):55–71, Nov. 1990.

[46] Alex F. Roche. Bone growth and maturation. In Frank Falkner and J.M. Tanner, editors, *Human Growth, Volume 2.* Plenum Press, 1978.

[47] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings),* volume 20, pages 151–160, August 1986.

[48] Gerard J. Tortora and Nicholas P. Anagnostakos. *Principles of Anatomy and Physiology.* Harper Collins, New York, 7th edition, 1993.

[49] Arthur J. Vander, James H. Sherman, and Dorothy S. Luciano. *Human Physiology.* McGraw-Hill Book Company, New York, third edition, 1980.

[50] Tsuneo Yoshikawa. *Foundations of Robotics.* MIT Press, Cambridge, MA, 1990.

[51] Jianmin Zhao. *Moving Posture Reconstruction from Perspective Projections of Jointed Figure Motion.* PhD thesis, University of Pennsylvania, 1993.